Emergent Behaviors of a Fuzzy Sensory-Motor Controller Evolved by Genetic Algorithm

Seung-Ik Lee, Sung-Bae Cho

Abstract

Recently, there has been extensive work on the construction of fuzzy controllers for mobile robots by genetic algorithm, so we can realize evolutionary optimization as a promising method for developing fuzzy controllers. However, much investigation on the evolutionary fuzzy controller remains because most of the previous works have not seriously attempted to analyze the fuzzy controller obtained by evolution. This paper develops a fuzzy logic controller for a mobile robot with a genetic algorithm in simulation environments and analyzes the behaviors of the controller with a state transition diagram of the internal model. Experimental results show that appropriate control mechanisms of the fuzzy controller are obtained by evolution. The controller has evolved well enough to smoothly drive the robot in different environments. The robot produces emergent behaviors by the interaction of several fuzzy rules obtained.

Keywords

Fuzzy controller, genetic algorithm, emergent behavior, mobile robot, Khepera

I. INTRODUCTION

SENSOR based mobile robot control has been actively and extensively studied so far. Researchers have applied several methodologies to obtaining more adaptive controllers because it is very important to adapt to continuously changing environments. Traditional approaches, driving the mobile robots after planning the path with internal representation of their surroundings, have the shortcoming in adapting them to changing environments. To overcome this problem, an approach called behavior-based or sensor-based method has been proposed. This approach takes the way of driving a mobile robot by direct mapping between sensors and motors without building predefined environmental maps [1].

Fuzzy logic controllers (FLC) [2]−[12] have been widely used in many application areas because they can easily transform linguistic information and expert knowledge into control signals. While fuzzy logic control has many advantages over traditional methods, it has also some drawbacks at the design stage in that it is difficult to acquire expert's knowledge and there are many parameters needed to be determined. Thus, many researchers have applied evolutionary algorithms to the construction of FLCs in order to automate the procedure of determining the parameters [5]−[10].

S.-I. Lee and S.-B. Cho are with the Dept. of Computer Science, Yonsei University, Seoul 120-749, South Korea. E-mail: cypher@candy.yonsei.ac.kr, sbcho@csai.yonsei.ac.kr.

This work was supported by Korea Research Foundation Grant (KRF 2000-005-C00012).

Fukuda [5] applied a genetic algorithm (GA) in the tuning of fuzzy membership functions to acquire the swimming motions of artificial fish. Beom [6] proposed a sensor-based method that utilized fuzzy logic and reinforcement learning for the navigation of a mobile robot in uncertain environments. In this work, fuzzy rule bases were built through the reinforcement learning that required simple evaluation data rather than input-output training data. Seng [7] proposed a neuro-fuzzy logic controller where all of its parameters could be tuned simultaneously by GA. The structure of the controller was based on the radial basis function neural network with Gaussian membership functions. They used a flexible position coding strategy of the neuro-fuzzy logic controller parameters to obtain near optimal solutions. Lim [8] suggested a paradigm for learning fuzzy rules using GA. They formulated their problem as follows: Given a set of linguistic values that characterize the input and output state variables of the system in consideration, derive an n-rule fuzzy control algorithm. They applied it to a classical inverted pendulum control problem to demonstrate the effectiveness of the GA learning scheme. Izumi [10] proposed a method to construct a fuzzy behavior-based control system when the robot cannot often receive any information from sensors by applying a virus evolutionary genetic algorithm with species [11] to generating the fuzzy rules.

The previous works show that the FLC parameters can be tuned to produce appropriate FLC rules, the shapes and centers of fuzzy membership functions by GA. Nonetheless, it is still necessary to reveal the control mechanisms of the FLC controller finally obtained by evolution so as to understand how the controller works for the robot.

In evolutionary biology and artificial life fields, there has been an implicit agreement that a complex and self-organized system shows emergent behaviors [13]−[16]. Generally, emergence can be understood as a process that leads to the appearance of structure not directly described by defining constraint and instantaneous forces to control a system. There are several definitions on the terminology of emergence. Morgan and Emmeche defined it as “the creation of new properties” in 1923 [13], [14]. Baas tried to define it more formally in a mathematical form [15].

In this paper, we have constructed the FLC whose internal parameters are tuned by GA to control the sensor-based mobile robot. In order to do this, we have represented the FLC parameters, the fuzzy sets and rules of the FLC, in genetic codes. We have analyzed the evolved control mechanism to show that the behaviors of the mobile robot emerge from the interaction among several primitive rules.

II. KHEPERA : MOBILE ROBOT

Khepera (see Fig. 1) was originally designed for research and teaching in the framework of a Swiss Research Priority Program. It allows confrontation to the real world of algorithms developed in simulation for trajectory execution, obstacle avoidance, pre-processing of sensory information, and hypothesis test on behavior processing.

A Khepera robot has two wheels. A DC motor coupled with the wheel through a 25:1 reduction gear moves every wheel. An incremental encoder is placed on the motor axis and gives 24 pulses per revolution of the motor. This allows a resolution of 600 pulses per revolution of the wheel that corresponds to 12 pulses per millimeter of path of the robot. The motor controller can be used in two control modes: speed and position modes. The active control mode is set

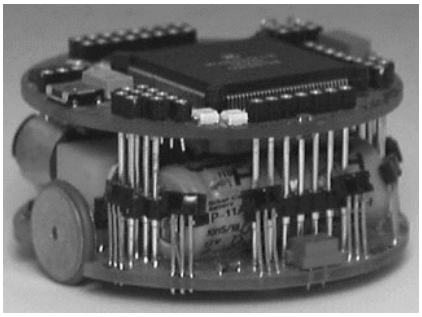


Fig. 1. Mobile robot, Khepera.

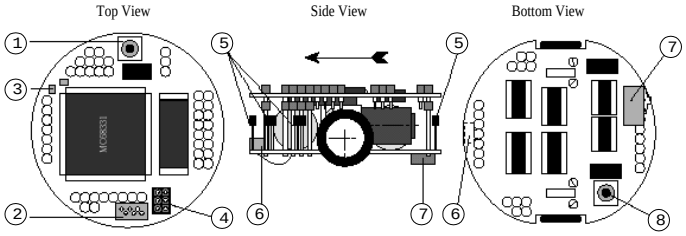


Fig. 2. Position of some parts of the robot [17]. ① LEDs. ② Serial line (S) connector. ③ Reset button. ④ Jumpers for the running mode selection. ⑤ Infra-Red proximity sensors. ⑥ Battery recharge connector. ⑦ ON-OFF battery switch. ⑧ Second reset button.

according to the kind of command received. If the controller receives a position control command, the control mode is automatically switched to the position mode. Used in speed mode, the controller has as input a speed value of the wheels, and controls the motor to keep this wheel speed. The speed modification is made as quick as possible, in an abrupt way. No limitation in acceleration is considered in this mode. Eight infrared proximity sensors are placed around the robot and are positioned and numbered as shown in Fig. 3. These sensors embed an infrared emitter and a receiver. This sensor device allows two measures:

- The normal ambient light. This measure is made using only the receiver part of the device, without emitting light with the emitter. A new measurement is made every 20ms. During the 20ms, the sensors are read in a sequential way every 2.5ms. The value returned at a given time is the result of the last measurement made.
- The light reflected by obstacles. This measure is made emitting light using the emitter part of the device. The returned value is the difference between the measurement made emitting light and the light measured without light emission (ambient light). A new measurement is made every 2.5ms. The value returned at a given time is the result of the last measurement made.

III. FUZZY LOGIC CONTROLLER

The basic structure of a fuzzy logic controller consists of three conceptual components: fuzzification of the input-output variables, a rule base which contains a set of fuzzy rules, and a reasoning mechanism which performs the inference procedure on the rules and given facts to derive a reasonable output. An example of a fuzzy inference system

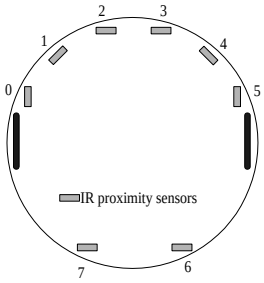


Fig. 3. Position of the 8 IR sensors.

with crisp output is shown in Fig. 4.

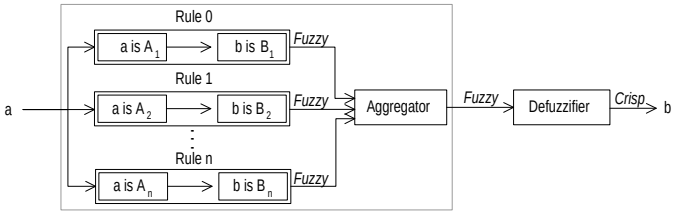


Fig. 4. Block diagram of a fuzzy logic controller.

A. Fuzzy Sets

Navigation of a mobile robot in uncertain environments requires avoiding obstacles. In order to reach a destination point without colliding into obstacles, a mobile robot should take sensory information about obstacles into account. Our FLC uses the sensory information of eight proximity sensors as inputs and controls the speed of the two motors on Khepera.

Let $d_i, i = 0, 1, \ldots, 7$, be input linguistic variables defined in the universe of discourse $U_d = [0, 1023]$ to represent the value of the ith proximity sensor of Khepera robot. A fuzzy set D_i on d_i is defined as follows:

$$D_i = \int_{U_d} \mu_{D_i}(d_i)/d_i \tag{1}$$

The integration sign in equation (1) stands for the union of $(d_i, \mu_{D_i}(d_i))$ pairs, and they do not indicate summation or integration. Similarly, "/" is only a marker and does not imply division.

In the same way, let $v_i, i = 0, 1$, be output linguistic variables defined in the universe of discourse $U_v = [-10, +10]$ to represent the speed of the ith motor of Khepera robot, and a fuzzy set V_i on v_i is defined as follows:

$$V_i = \int_{U_v} \mu_{V_i}(v_i)/v_i \tag{2}$$

The input linguistic variable d_i and output linguistic variable v_i are expressed by linguistic values (VF, F, C, VC) and (BH, B, F, FH) respectively. The linguistic values have the meanings as in table I.

The membership functions of D_i and V_i are all in a triangular form defined by equation (3).

TABLE I

Linguistic values and meanings.

Terms	Meaning	Terms	Meaning
VF	Very Far	BH	Backward High
F	Far	B	Backward
C	Close	F	Forward
VC	Very Close	FH	Forward High

$$\text{triangle}(x, a, b, c) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{3}$$

where the parameters $\{a, b, c\}$ with $a < b < c$ determine the x coordinates of the three corners of the underlying triangular membership function. To reduce the computational complexity, some restrictions are applied to each membership function as in the following equations (see also Fig. 5).

$$\begin{aligned}
\mu_{d_i,VF}(d_i) &= \text{triangle}(0, 0, b_{d_i}^{a_i}) \\
\mu_{d_i,F}(d_i) &= \text{triangle}(0, b_{d_i}^{a_i}, b_{d_i}^{b_i}) \\
\mu_{d_i,C}(d_i) &= \text{triangle}(b_{d_i}^{a_i}, b_{d_i}^{b_i}, 1023) \\
\mu_{d_i,VC}(d_i) &= \text{triangle}(b_{d_i}^{b_i}, 1023, 1023) \\
\mu_{v_i,BH}(v_i) &= \text{triangle}(-10, -10, b_{v_i}^{c_i}) \\
\mu_{v_i,B}(v_i) &= \text{triangle}(-10, b_{v_i}^{c_i}, b_{v_i}^{d_i}) \\
\mu_{v_i,F}(v_i) &= \text{triangle}(b_{v_i}^{c_i}, b_{v_i}^{d_i}, +10) \\
\mu_{v_i,FH}(v_i) &= \text{triangle}(b_{v_i}^{d_i}, +10, +10)
\end{aligned} \tag{4}$$

where $0 < b_{d_i}^{a_i} < b_{d_i}^{b_i} < 1023$ and $-10 < b_{v_i}^{c_i} < b_{v_i}^{d_i} < +10$. $b_{d_i}^{a_i}$ and $b_{d_i}^{b_i}$ are two of $b_{d_i}^{k}, k = 0, \ldots, 9$, evenly distributed in the universe of discourse U_d, while $b_{v_i}^{c_i}$ and $\text{b}_{v_i}^{d_i}$ are two of $b_{v_i}^{k}, k = 0, \ldots, 9$, evenly distributed in the universe of discourse U_v.

B. Fuzzy Rule Base

Generally, a human expert can construct the rule base of an FLC based on their experience. However, when it comes to more complex work, it is difficult to tune the rule base with only on human expertise. To overcome this difficulty, we have used a genetic algorithm as an automatic way to optimize the rule base. The rule base for the FLC consists of the rules of IF-THEN statements as follows.

R_0 : IF ($d_0 = D_0^0$) and $\cdots$ and ($d_7 = D_7^0$)
THEN ($v_0 = V_0^0$) and ($v_1 = V_1^0$)

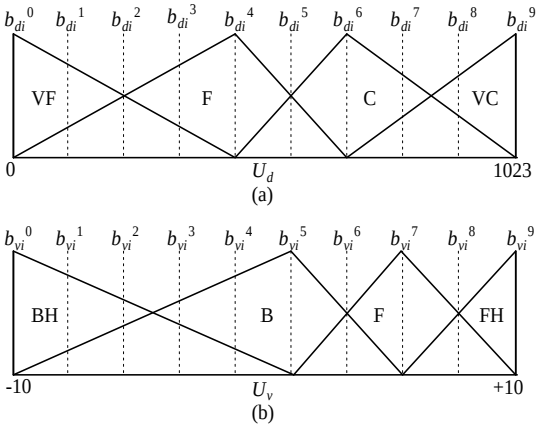


Fig. 5. The membership functions of input (a) and output (b) variables.

$$\vdots$$

R_n : IF $(d_0 = D_0^n)$ and $\cdots$ and $(d_7 = D_7^n)$
THEN $(v_0 = V_0^n)$ and $(v_1 = V_1^n)$

$$\vdots$$

R_{N-1} : IF $(d_0 = D_0^{N-1})$ and $\cdots$ and $(d_7 = D_7^{N-1})$
THEN $(v_0 = V_0^{N-1})$ and $(v_1 = V_1^{N-1})$

where R_n denotes the nth rule. Also, D_i^n and V_i^n, $n = 0, 1, \ldots, N-1$, are the fuzzy sets defined on d_i and v_i in the universe of discourse U_d and U_v respectively. Here, N denotes the number of the rules in FLC. The nth rule can be represented as a fuzzy relation defined by equation (5).

$$\begin{aligned} R_n :&(D_0^n \times D_1^n \times D_2^n \times D_3^n \times D_4^n \times D_5^n \times D_6^n \times D_7^n) \\ &\rightarrow (V_0^n, V_1^n) \end{aligned} \tag{5}$$

where $\rightarrow$ denotes fuzzy relation. The rule base for the behaviors of mobile robot can be represented as union as follows:

$$\begin{aligned} R &= \left\{ \bigcup_{n=0}^{N-1} R_n \right\} \\ &= \left\{ \bigcup_{n=0}^{N-1} (D_0^n \times D_1^n \times \cdots \times D_7^n) \rightarrow (V_0^n, V_1^n) \right\} \\ &= \left\{ \bigcup_{n=0}^{N-1} \left[(D_0^n \times D_1^n \times \cdots \times D_7^n) \rightarrow V_0^n, (D_0^n \times D_1^n \times \cdots \times D_7^n) \rightarrow V_1^n \right] \right\} \\ &= \left\{ \bigcup_{n=0}^{N-1} RV_0^n, \bigcup_{n=0}^{N-1} RV_1^n \right\} = \{RV_0, RV_1\} \end{aligned} \tag{6}$$

where R consists of two sub rule bases RV_0 and RV_1 which control the speed of mobile robot. Each rule base consists of N rules, each of which can be represented by a fuzzy relation in the product space expressed by equation (7).

$$U_d \times U_d \times U_d \times U_d \times U_d \times U_d \times U_d \times U_v \tag{7}$$

This fuzzy relation can be implemented with each corresponding membership function defined by equations (8) and (9).

$$\begin{aligned}&\mu_{RV_0^n}(d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, v_0)\\&\quad = f(\mu_{D_0^n}(d_0), \cdots, \mu_{D_7^n}(d_7), \mu_{V_0^n}(v_0))\end{aligned} \tag{8}$$

$$\begin{aligned}&\mu_{RV_1^n}(d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, v_1)\\&\quad = f(\mu_{D_0^n}(d_0), \cdots, \mu_{D_7^n}(d_7), \mu_{V_1^n}(v_1))\end{aligned} \tag{9}$$

C. Fuzzy Reasoning

Let D_i^n and $D_i^{n'}$ be the fuzzy sets defined on d_i in the universe of discourse U_d and $V_0^n, V_1^n, V_0^{n'}$ and $V_1^{n'}$ be the fuzzy sets defined on v_0 and v_1 in the universe of discourse U_v respectively. To control the actions of mobile robot, $V_0^{n'}$ and $V_1^{n'}$ should be inferred from $D_i^n, D_i^{n'}, V_0^n$ and V_1^n. The nth rule R_n can be transformed into a fuzzy relation based on Mamdani's fuzzy implication function [18] as shown in the following equations.

$$\begin{aligned}&RV_0^n(D_0^n, \cdots, D_7^n, V_0^n) = (D_0^n \times \cdots \times D_7^n) \times V_0^n\\&= \int_{U_d \times \cdots \times U_d \times U_v} \mu_{D_0^n}(d_0) \wedge \cdots \wedge \mu_{D_7^n}(d_7) \wedge \mu_{V_0^n}(v_0)/(d_0, \cdots, d_7, v_0)\end{aligned} \tag{10}$$

$$\begin{aligned}&RV_1^n(D_0^n, \cdots, D_7^n, V_1^n) = (D_0^n \times \cdots \times D_7^n) \times V_1^n\\&= \int_{U_d \times \cdots \times U_d \times U_v} \mu_{D_0^n}(d_0) \wedge \cdots \wedge \mu_{D_7^n}(d_7) \wedge \mu_{V_1^n}(v_1)/(d_0, \cdots, d_7, v_1)\end{aligned} \tag{11}$$

Based on Zadeh's compositional rule of inference [19], V_0' and V_1' are expressed as

$$\begin{aligned}V_0' &= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} (D_0^n \times \cdots \times D_7^n \to V_0^n)\\&= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} RV_0^n\end{aligned} \tag{12}$$

$$\begin{aligned}V_1' &= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} (D_0^n \times \cdots \times D_7^n \to V_1^n)\\&= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} RV_1^n\end{aligned} \tag{13}$$

where $\circ$ denotes the maximum-minimum composition. The resulting V_0' and V_1' are expressed as in the following equations.

$$\begin{aligned}
\mu_{V_0'} &= \bigcup_{n=0}^{N-1} \vee_{d_0,\cdots,d_7} \left\{ \left[\mu_{D_0^{n'}}(d_0) \wedge \cdots \wedge \mu_{D_7^{n'}}(d_7)\right] \wedge \left[\mu_{D_0^n}(d_0) \wedge \cdots \wedge \mu_{D_7^n}(d_7) \wedge \mu_{V_0^n}(v_0)\right] \right\} \\
&= \bigcup_{n=0}^{N-1} \vee_{d_0,\cdots,d_7} \left\{ \left[\mu_{D_0^{n'}}(d_0) \wedge \cdots \wedge \mu_{D_7^{n'}}(d_7) \wedge \mu_{D_0^n}(d_0) \wedge \cdots \wedge \mu_{D_7^n}(d_7)\right] \right\} \wedge \mu_{V_0^n}(v_0) \\
&= \bigcup_{n=0}^{N-1} \underbrace{\left\{ \vee_{d_0} \left[\mu_{D_0^{n'}}(d_0) \wedge \mu_{D_0^n}(d_0)\right] \right\}}_{\omega_0} \wedge \cdots \wedge \underbrace{\left\{ \vee_{d_7} \left[\mu_{D_7^{n'}}(d_7) \wedge \mu_{D_7^n}(d_7)\right] \right\}}_{\omega_7} \wedge \mu_{V_0^n}(v_0) \\
&= \bigcup_{n=0}^{N-1} \underbrace{(\omega_0 \wedge \cdots \wedge \omega_7)}_{\text{firing strength}} \wedge \mu_{V_0^n}(v_0)
\end{aligned} \tag{14}$$

Similarly, $\mu_{V_1'}$ is defined as

$$\mu_{V_1'} = \bigcup_{n=0}^{N-1} \underbrace{(\omega_0 \wedge \cdots \wedge \omega_7)}_{\text{firing strength}} \wedge \mu_{V_1^n}(v_1) \tag{15}$$

where $\wedge$ denotes the minimum operation and ω_i is the maxima of the membership functions of $D_i^n \cap D_i^{n'}$.

D. Defuzzification

Defuzzification refers to the way $\bar{v}_0$ and $\bar{v}_1$ are extracted from a fuzzy set as representative values. Among many defuzzification methods [18], [20]$-$[22], the center of gravity method is used because it is widely used and also appropriate for our system to control the mobile robot.

$$\bar{v}_0 = \frac{\int_{v_0} \mu_{v_0}(v_0) v_0 dv_0}{\int_{v_0} \mu_{v_0}(v_0) dv_0} \tag{16}$$

$$\bar{v}_1 = \frac{\int_{v_1} \mu_{v_1}(v_1) v_1 dv_1}{\int_{v_1} \mu_{v_1}(v_1) dv_1} \tag{17}$$

IV. EVOLUTION OF FLC PARAMETERS

A genetic algorithm (GA) is a search technique based on the mechanics of natural selection and natural genetics [23]. This combines survival of the fittest among string structures with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In every generation, a new set of strings is created using bits and pieces of the fittest of the old; an occasional new part is tried for good measure. While randomized, genetic algorithms are no simple random walk. They efficiently exploit historical information to speculate on new search points with expected improved performance. The general procedure of GA is shown in Fig. 6.

At first, a population of individuals that encode candidate solutions to given problem is initialized at random. Each individual in the population is evaluated in the problem at hand and changed by genetic operations such as crossover and mutation to reproduce a new population. This process goes on until a satisfactory individual appears in the population.

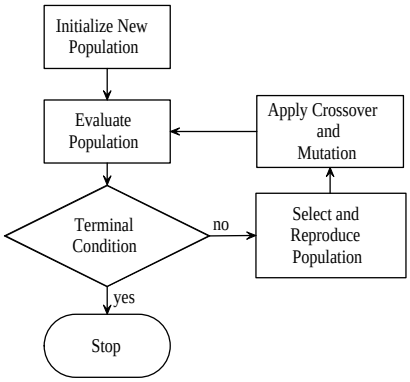


Fig. 6. General procedure of GA.

The whole process of tuning FLC parameters using GA is depicted in Fig. 7. The code of an individual representing the parameters of FLC is applied to the Khepera robot in the environment to measure the fitness. After some period of time, each individual FLC is given a fitness value according to its performance in a given domain. The individuals with higher fitness are selected and go through genetic operations to produce the next population of individuals. In the example of Fig. 7, the 1st, 2nd, 5th, and 7th individuals are selected because they have higher fitness values than the others. This whole process continues until a controller showing good performance in driving the robot in the environment is found.

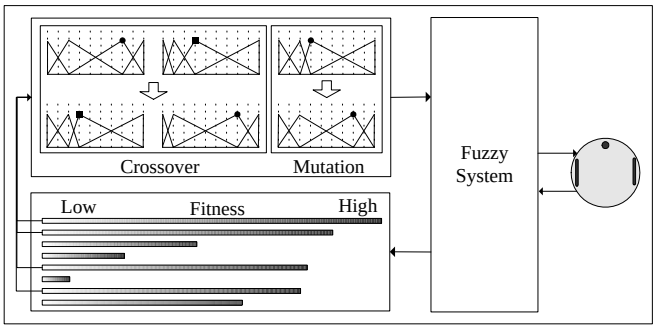


Fig. 7. Fuzzy system tuning by GA.

There are two parameters that should be determined to run GA: how to encode the FLC parameters in gene code and how to estimate the fitness value of each individual. For the FLC parameters, eight input variables, two output variables, and maximally ten rules are encoded as shown in Fig. 8.

8 INPUTS	2 OUTPUTS	10 RULES

Fig. 8. Encoding of FLC parameters.

Sixty bits are required to encode all the fuzzy sets defined on all the input-output variables because only two of the four fuzzy sets of a variable need to be encoded (see equation (4)) and only six bits are required to encode two fuzzy sets (see Fig. 9).

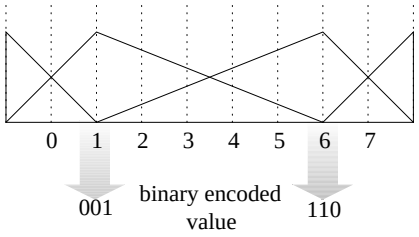


Fig. 9. Encoding of membership function.

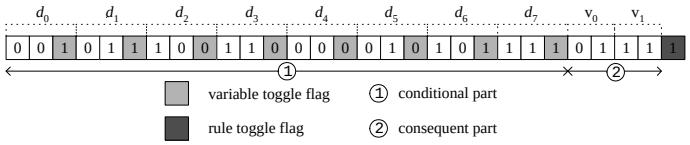


Fig. 10. Encoding of a rule.

Each rule has eight input variables, $d_0, \ldots, d_7$, and two output variables, v_0 and v_1. Three bits represent each input variable. Two of them (white cells in Fig. 10) are used for coding one of the four fuzzy sets, VF (00), F (01), C (10), and VC (11). The last one is a toggle bit. The variable having the toggle bit 1 participates in the conditional part in the fuzzy rule. In output variables, the two bits of each variable are used for coding one of the four fuzzy sets, BH (00), B (01), F (10), FH (11). There are no toggle flags because all the output variables should appear in consequent part. Finally, the last bit in Fig. 10 designates whether this rule participates in fuzzy inference process or not. Therefore, Fig. 10 can be decoded as follows:

$$\begin{aligned} &\text{IF } (d_0 = \text{VF}) \text{ and } (d_1 = \text{F}) \text{ and } (d_6 = \text{C}) \text{ and } (d_7 = \text{VC}) \\ &\quad \text{THEN } (v_0 = \text{B}) \text{ and } (v_1 = \text{FH}) \end{aligned}$$

To use GA for the evolution of individuals representing FLC for the mobile robot, a fitness function is defined by equation (18).

$$\begin{aligned} f =& \alpha(\text{number of collisions}) + \beta(\text{moving distance}) + \\ & \gamma(\text{number of rules}) + \delta(\text{number of fuzzy sets}) + \\ & \epsilon(\text{check points}) \end{aligned} \tag{18}$$

where $\alpha = -3,\ \beta = 1,\ \gamma = -100,\ \delta = -10,$ and $\epsilon = 500$. Here, only 'moving distance' and 'check points' have positive effects on the function f. Therefore, the robot that moves long distance or passes through many check points gets higher fitness value. On the other hand, the robot that collides against the wall or has many rules and fuzzy sets gets lower fitness value.

V. SIMULATION RESULTS

Experiments have been performed on SUN SparcStation 10 machine. At the beginning of evolution, two hundred individuals are initialized at random. Crossover and mutation operations are applied with the probability of 0.5 and 0.05,

respectively. Then each individual is decoded into an FLC, which controls the Khepera mobile robot in a simulation environment like Fig. 11. Each individual is evaluated for five thousands times of sensor sampling.

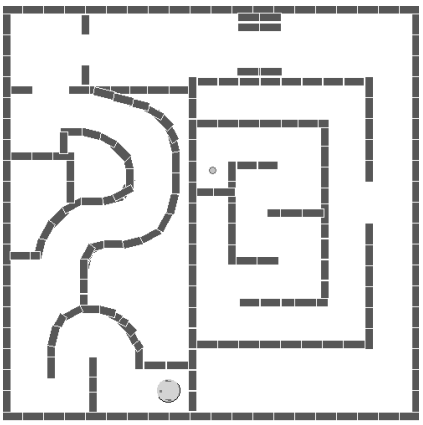


Fig. 11. The environment used for evolving the robot.

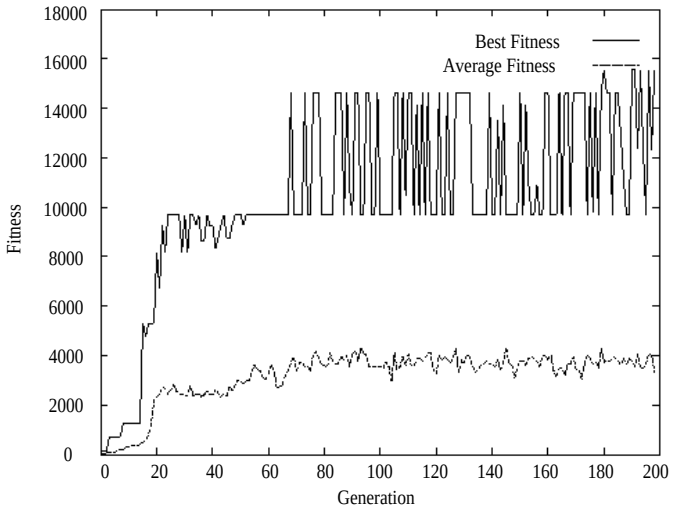


Fig. 12. Fitness change by generation.

Fig. 12 shows the fitness change over generations. In early generations, the fitness has radically increased with some oscillation. After the 67th generation, the best fitness does not increase significantly. On the contrary, average fitness steadily and continuously increases as each generation passes. Although we have used an elite preserving strategy in the selection process [23], the best fitness value oscillates because the sensor information is noisy to make the simulation more realistic. From the 64th generation, individuals with extremely high fitness value start to show up and disappear. Among the individuals, the first individual that reaches the goal position has appeared from the 67th to 70th generations. The movement of the best individual is shown in Fig. 13. The robot can turn right, left, and around even though we did not give any information on the parameters to the fuzzy system.

The fuzzy rules of the best individual are shown in Table II, and Fig. 14 shows the associated fuzzy sets.

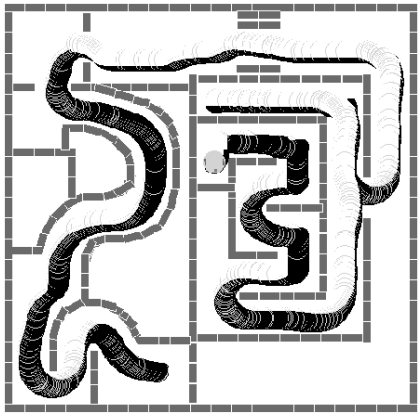


Fig. 13. Trajectory of a successful robot in the simulation environment.

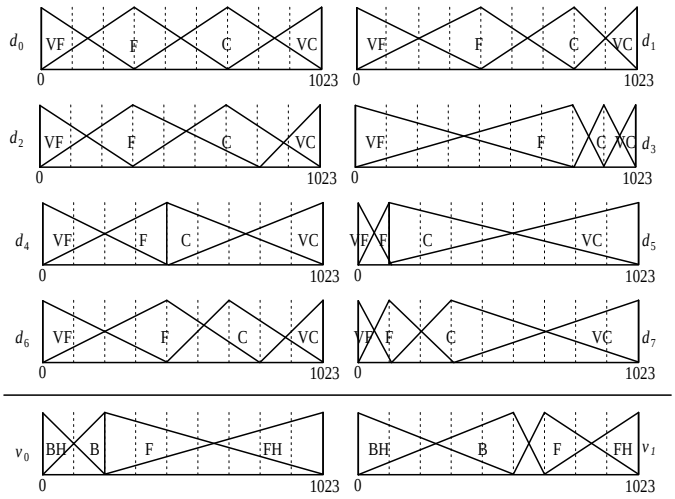


Fig. 14. Fuzzy sets obtained by evolution.

VI. ANALYSIS

The robot has implicitly built an internal behavior model of three states, each of which has its own action to survive until it reaches the target point.

The behavior model can be described as shown in Fig. 15. 'Corridor Passing' state is activated when the firing strength of rule 2 is greater than zero. This state controls the robot when it is passing through narrow path, 'Corridor', surrounded by wall on both sides. 'Wall Following' is activated when the firing strengths of rule 2 and 7 are greater than zero. 'Wall Following' controls the robot when it is in a situation where a wall on right side and not on the other side. 'Collision Avoidance' is activated when the firing strength of rule 5 is greater than zero. This state controls the robot when it is confronted a obstacle in the front.

As you can see in Table II, the rules obtained by evolution produce appropriate actions of the robot against corresponding sensory information. But to get to the goal point, the robot has to be able to do higher-level behaviors such as turning left, turning right, and turning around that are not explicitly described in the rules in Table II. As shown in Fig. 13, even when the robot confronts the situations where each rule cannot produce the higher behavior, it makes

TABLE II

Evolved fuzzy rules.

Rule 1 : IF (d_2 =C) and (d_5 =VF) and (d_7 =VC)
THEN (v_0 =BH) and (v_1 =B)

Rule 2 : IF (d_4 =VF)
THEN (v_0 =FH) and (v_1 =F)

Rule 3 : IF (d_1 =VC) and (d_2 =F) and (d_4 =C) and (d_7 =VC)
THEN (v_0 =BH) and (v_1 =B)

Rule 4 : IF (d_2 =F) and (d_3 =F) and (d_6 =VC)
THEN (v_0 =F) and (v_1 =FH)

Rule 5 : IF (d_4 =VC)
THEN (v_0 =BH) and (v_1 =F)

Rule 6 : IF (d_2 =VF) and (d_4 =F) and (d_6 =VC)
THEN (v_0 =F) and (v_1 =FH)

Rule 7 : IF (d_0 =VF) and (d_4 =F) and (d_5 =C)
THEN (v_0 =BH) and (v_1 =F)

appropriate action through cooperation of the rules using the behavior model described in Fig. 15. Although we did not incorporate any a priori knowledge into the rules that are appropriate for the higher-level behaviors, the appropriate behaviors emerge based on the cooperation of the rules in Table II, which are very similar to the emergent behaviors from the notion of "the creation of new properties" [13], [14].

A. Passing through Corridor

Fig. 16 shows the trajectories of the robot while it is passing through one of corridors in our experiment environments. A corridor can be curved or straight as in Fig. 16. From step 1 to step 97, the robot is in 'Corridor Passing' state using rule 2. Fig. 17 shows some data obtained when the robot moves forward passing through the corridor: (a) shows the speed of the robot with respect to the activation level of rule 2. As the activation level of rule 2 increases, the behaviors of the robot change from 'turning left' to 'moving forward' and then to 'turning right'. 'Turning left' behavior comes when the activation level of rule 2 is less than or equal to 0.23 because the speed of the right motor remains at 5 and the speed of the left motor remains below that of the right motor as can be seen in Fig. 17 (a). When the activation level is in 0.23$\sim$0.35, two motors have the same speed so that the robot moves forward, and when the activation is greater than 0.35, the robot does the 'turning right' behavior.

In Fig. 16, the robot seems to move straight forward. This implies that the activation level of rule 2 should be in

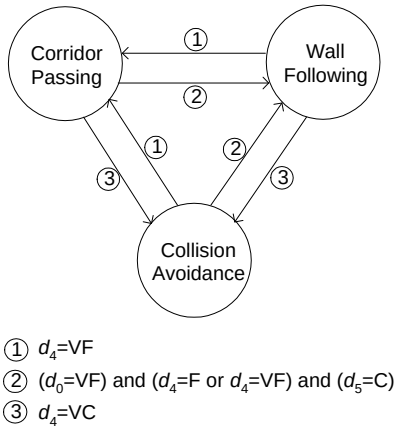


Fig. 15. Behavior model of the robot.

Fig. 16. Passing through corridor: (a) Step 1 (b) Step 6 (c) Step 14 (d) Step 97.

0.23∼0.35 (see Fig. 17 (a)). But as can be seen in Fig. 17, the sensor values of d_4 oscillate with wide range. As a result, the activation level of rule 2 also oscillates from 0.025 to 0.425. With these activation levels, the robot indeed does the toggling behaviors of 'turning left' and 'turning right' which in combination lead to moving forward. Therefore, although the robot looks like 'moving straight forward' behavior in Fig. 16, it actually does 'turning left' and 'turning right' behaviors (see Fig. 17 (d)).

In summary, when the robot approaches to the right wall (high d_4 sensor value $\rightarrow$ low activation level of rule 2) the robot turns left, and when it approaches to the left wall (low d_4 sensor value $\rightarrow$ high activation level of rule 2) the robot turns right. With these behaviors, the robot moves forward. This implies that the 'Corridor Passing' state in Fig. 15 contains two substates as in Fig. 18.

At step 1 in Fig. 17 (c), the activation level of rule 2 is about 0.425 and the robot is in 'Turn Right' state. When the robot is in this state, it turns right with the right motor of speed 4 and the left motor of speed 5 (see Fig. 17 (d)). However, when the activation level of rule 2 becomes less than or equal to 0.35 like step 6 in Fig. 17 (c), the robot changes its state from 'Turn Right' to 'Turn Left' and finally turns left away from the right obstacle with the left speed of 2 and the right speed of 5. Therefore, the behaviors shown in Fig. 16 are the result of changing the two substates, 'Turn Right' and 'Turn Left'.

B. Turning Around

Fig. 19 shows a situation where the robot collides against a wall so that it can go forward no more and should turn around to the opposite direction. This behavior is also important for the robot to reach the goal position. During the steps from 1 to 61 in Fig. 19, the robot uses all the states shown in Fig. 15. Especially when the robot reaches the obstacle

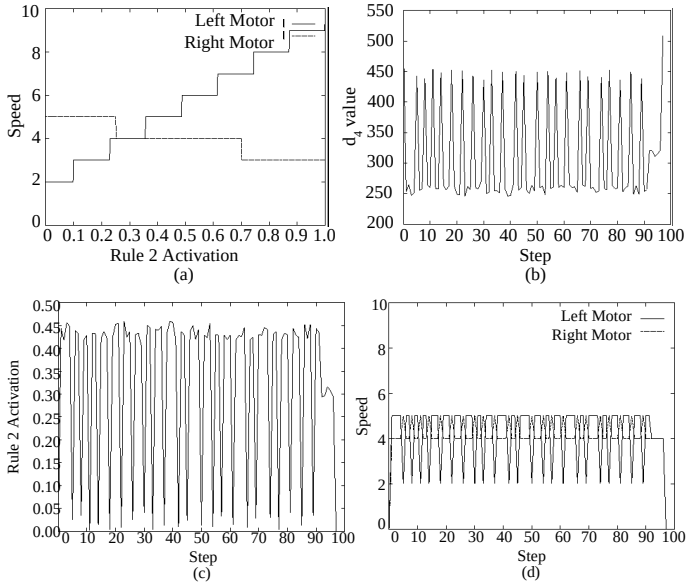


Fig. 17. Data obtained when the robot passes through a corridor: (a) Speed according to the activation of rule 2 (b) d_4 sensor values (c) Rule 2 activation (d) Stepwise speed change of the robot.

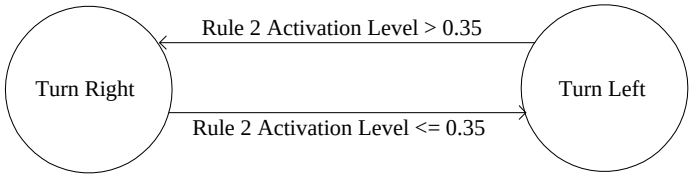


Fig. 18. State interleaving for the behaviors of 'Corridor Passing'.

wall, it mainly uses 'Collision Avoidance' and 'Wall Following' states. The former is activated by the activation of rule 5 and the latter is activated by the activation of rules 2 and 7.

Fig. 19. Turning around: (a) Step 1 (b) Step 31 (c) Step 41 (d) Step 61.

Fig. 20 shows some data obtained when the robot turns around: Fig. 20 (a) shows the speed of the robot according to the activation level of rule 5. As can be seen, whenever the rule 5 is activated, the robot turns left. Fig. 20 (b) shows the sensing values of the related sensors while the robot turns around from step 1 to step 100. Fig. 20 (c) shows the activation levels of related rules, 2, 5, and 7, and Fig. 20 (d) shows the speed changes of two motors during this process.

We can infer from the radical changes of the sensing values from step 31 to 60 in Fig. 20 (b) that the robot gets close to the front wall. The activation of related rules (Fig. 20 (c)) and speed (Fig. 20 (d)) are also changing dramatically just like the sensing values. The state changes from step 1 to step 100 are shown in Fig. 21. Steps before 31 are governed by 'Corridor Passing' state which causes the robot to avoid colliding to left and right walls, and the steps from 31 to 61 are

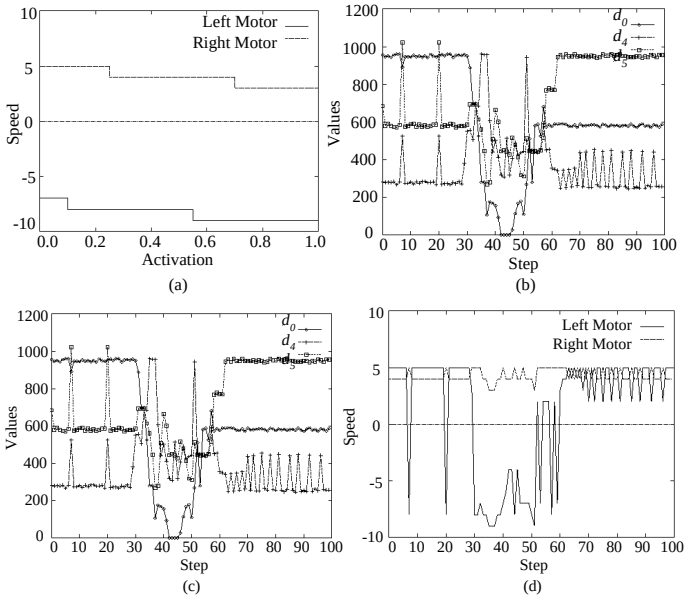


Fig. 20. Data obtained when the robot turns around: (a) Speed according to the activation of rule 5 (b) d_0, d_4, and d_5 sensor values (c) Activation of related rules (d) Stepwise speed change of the robot.

governed by ‘Collision Avoidance’ and ‘Wall Following’ states which cause the robot to turn around without colliding to the front wall.

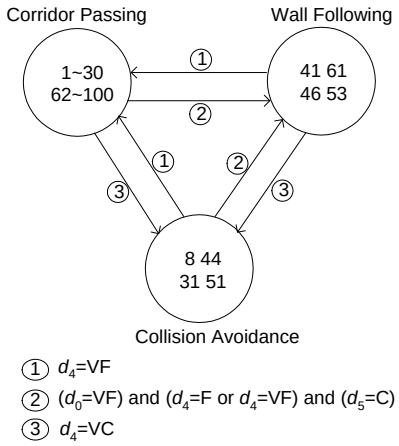


Fig. 21. Internal state transition of the robot when turning around.

C. Cornering

Fig. 22 shows a situation where the robot turns a corner. During the steps from 1 to 90, the robot uses two states, ‘Corridor Passing’ and ‘Wall Following’ (see Fig. 23). From step 1 to 57, the robot uses all the two states, while it uses only ‘Corridor Passing’ state after step 58.

Fig. 24 shows some data obtained when the robot turns the corner. Fig. 24 (a) shows the activation levels of related rules, 2 and 7, and Fig. 24 (b) shows the speed changes of two motors during this process. With the cooperation of the

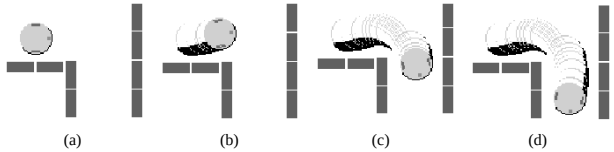


Fig. 22. Cornering: (a) Step 1 (b) Step 28 (c) Step 68 (d) Step 90

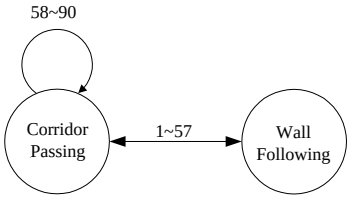


Fig. 23. Internal state transition of the robot during cornering.

two states, the robot smoothly turns the corner.

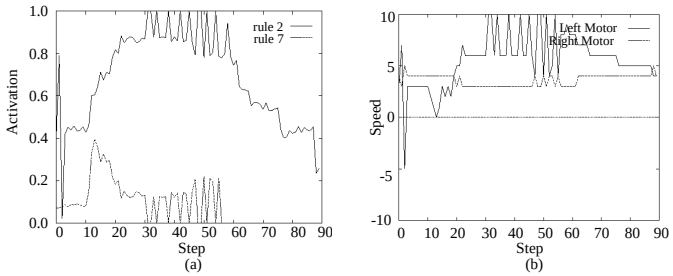


Fig. 24. Data obtained when the robot turns corner: (a) Activation of related rules (b) Stepwise speed change of the robot.

Without the cooperation of the two states, the robot fails to turn the corner or collides to the corner. Fig. 25 shows what happens when there is no cooperation of the two states. As can be seen, the robot collides to the corner in case of using only 'Corridor Passing' state. Therefore, we can assert that the robot has acquired the ability of cooperation among basis rules during the evolution process.

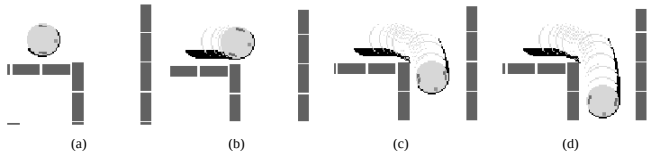
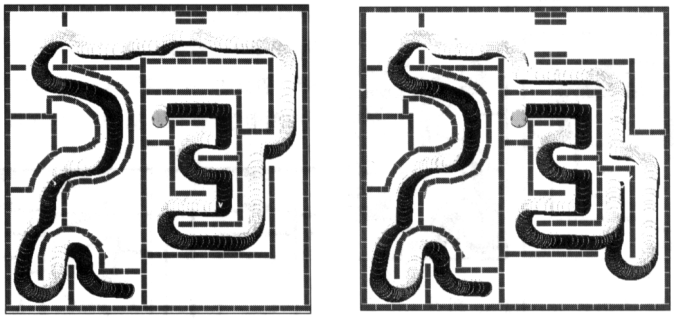


Fig. 25. Cornering using 'Corridor Passing' state: (a) Step 1 (b) Step 28 (c) Step 68 (d) Step 90.

D. Adaptive Behaviors in Different Environments

As can be seen in the previous sections, the evolved FLC can control the Khepera mobile robot appropriately in a given environment. In this section, application to different environments is presented. The environments are different from the trained environment as shown in Fig. 26.



(a) (b)

Fig. 26. Adaptation to new environments: (a) New environment 1 (b) New environment 2.

In these new environments, the robot is also able to reach the goal point, which means that the evolved FLC has the ability to navigate in environments that are different from the trained. The problems that have the same level of complexity as Fig. 26 might be solved and our approach is promising in that the controller is automatically constructed through the adaptation to the environments autonomously.

VII. CONCLUSION

In this paper, we have analyzed the FLC of a mobile robot evolved by a GA with a state transition diagram. Theoretically there can be 2^{28} rules for there are eight input variables and two output variables in a rule and four fuzzy sets per variable and a toggle flag with every input variable, but the FLC finally constructed by evolution consists of just seven rules with one to three input variables and even more only three of them are used to get to the goal position. This indicates that the input variables are properly optimized.

The robot has obtained proper mechanisms for several behaviors like turning left, turning right, and turning around to navigate in the complex environment. Using only one rule it can perform some of the behaviors while more sophisticated behaviors can be performed by the cooperation of several rules. The robot develops an internal model for behaviors implicitly represented by the rules. We have used this model to explain the behaviors of the mobile robot in several situations.

It is not desirable to limit the number of rules of the FLC, but acceptable because of time and efficiency. The simulation results also show that the limitation of the number of rules of the FLC did not affect the exploration of optimized solution because the final FLC consisted of only seven rules and could control the mobile robot appropriately.

As further research, the fitness function should be revised to contain more components such as wheel torques and velocities in order for the robot to turn around without tumbling over in real environments. Real world experiments should be also done to support the simulation results. Comparisons are necessary on the performance and adaptability with other conventional methods for constructing mobile robot controllers.

References

[1] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Transactions on Robotics and Automaton*, vol. 2, no. 1, pp. 14-23, 1986.

[2] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338-353, 1965.

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

[4] P. J. King and E. H. Mamdani, "The application of fuzzy control system to industrial process," *Automatica*, vol. 13, pp. 235-242, 1977.

[5] T. Fukuda, A. Kawamoto, and K. Shimojima, "Acquisition of swimming motion by RBF fuzzy neuro with unsupervised learning," *Proceedings of the 1995 International Workshop on Biologically Inspired Evolutionary Systems*, pp. 118-123, 1995.

[6] H. R. Beom and H. S. Cho, "A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 3, pp. 464-477, 1995.

[7] T. L. Seng, M. B. Khalid, and R. Yusof, "Tuning of a neuro-fuzzy controller by genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics-Part b: Cybernetics*, vol. 29, no. 2, pp. 226-236, 1999.

[8] M. H. Lim, S. Rahardja, and B. W. Gwee, "A GA paradigm for learning fuzzy rules," *Fuzzy Sets and Systems*, vol. 82, pp. 177-186, 1996.

[9] S.-B. Cho and S.-I. Lee, "Mobile robot learning by evolution of fuzzy controller," *Journal of Intelligent and Fuzzy Systems*, vol. 6, pp. 91-97, 1997.

[10] K. Izumi, K. Watanabe, T. Miyazaki, "Fuzzy behavior-based control for a miniature mobile robot," *Proceesings of Knowledge-Based Electronic Systems*, vol. 3, pp. 483-490, 1998.

[11] K. Watanabe, T. Miyazaki, K. Izumi, "Virus evolutionary genetic algorithm with species," *Proceesings of the 96th SICE Annual Conference*, vol. 1, pp. 447-448, 1997.

[12] C. C. Lee, "Fuzzy logic in control systems : Fuzzy logic controller, part 2," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 419-435, 1990.

[13] C. L. Morgan, *Emergent Evolution*, Williams and Norgate, 1923.

[14] C. Emmeche, S. Koppe, and F. Stjernfelt, "Explaining emergence: Towards an ontology of levels," *Journal for General Philosophy of Science*, vol. 28, pp. 83-119, 1997.

[15] N. A. Baas, "Emergence, hierarchies, and hyperstructures," *Artificial Life 3*, pp. 515-537, 1992.

[16] J. P. Crutchfield, "Is anything new?," *Sciences of Complexity XIX*, Addision-Wesley, 1994.

[17] K-Team, *Khepera Simulator Version 5.02 User Manual*, 1999.

[18] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.

[19] L. A. Zadeh, "Outline of a new approach to the analysis of complex systems and decision processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, no. 1, pp. 28-44, 1973.

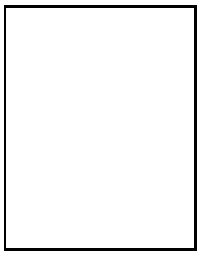
[20] N. Pfluger, J. Yen, and R. Langari, "A defuzzification strategy for a fuzzy logic controller employing prohibitive information in command formulation," *Proceedings of IEEE International Conference on Fuzzy Systems*, pp. 717-723, 1992.

[21] M. Sugeno and G. T. Kang, "Structure identification of fuzzy model," *Fuzzy Sets and Systems*, vol. 28, 1994.

[22] R. R. Yager and D. P. Filev, "A simple adaptive defuzzification method," *IEEE Transactions on Fuzzy Systems*, vol. 1, no. 1, pp. 69-78, 1993.

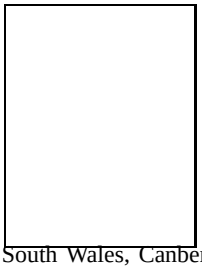
[23] D. E. Goldberg, *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison Wesley, 1989.

CAPTION LIST



Seung-Ik Lee received the B.S. and M.S. degrees in computer science from Yonsei University, Seoul, South Korea, in 1995 and 1997, respectively.

He is in a doctorial course at Yonsei University. He has published several papers on fuzzy logic control, evolutionary learning, and conversational agent. His research interests include evolutionary computation, fuzzy logic, conversational agent.



Sung-Bae Cho received the B.S. degree in computer science from Yonsei University, Seoul, Korea, in 1988 and the M.S. and Ph.D. degrees in computer science from KAIST (Korea Advanced Institute of Science and Technology), Taejeon, Korea, in 1990 and 1993, respectively.

He worked as a Member of the Research Staff at the Center for Artificial Intelligence Research at KAIST from 1991 to 1993. He was an Invited Researcher of Human Information Processing Research Laboratories at ATR (Advanced Telecommunications Research) Institute, Kyoto, Japan from 1993 to 1995, and a Visiting Scholar at University of New South Wales, Canberra, Australia in 1998. Since 1995, he has been an Associate Professor in the Department of Computer Science, Yonsei University. His researhc interests include neural networks, pattern recognition, intelligent man-machine interfaces, evolutionary computation, and artificial life.

Dr. Cho was awarded outstanding paper prizes from the IEEE Korea Section in 1989 and 1992, and another one from the Korea Information Science Society in 1990. He was also the recipient of the Richard E. Merwin prize from the IEEE Computer Society in 1993. He was listed in Who's Who in Pattern Recognition from the International Association for Pattern Recognition in 1994, and received the best paper awards at International Conference on Soft Computing in 1996 and 1998. Also, he received the best paper award at World Automation Congress in 1998, and listed in Marquis Who's Who in Science and Engineering in 2000 and in Marquis Who's Who in the World in 2001. He is a Member of the Korea Information Science Society, INNS, the IEEE Computer Society, and the IEEE Systems, Man, and Cybernetics Society.