Observational Emergence of a Fuzzy Controller Evolved by Genetic Algorithm

Seung-Ik Lee
Dept. of Computer Science, Yonsei University
Seoul 120-749, South Korea
cypher@candy.yonsei.ac.kr

Sung-Bae Cho
Dept. of Computer Science, Yonsei University
Seoul 120-749, South Korea
sbcho@csai.yonsei.ac.kr

Abstract- Explaining emergence is a difficult work such that there are many arguments on what it is or how it can be explained. Nonetheless, it is frequently referred in many fields such as behavior based robotics, artificial life, and complex systems without any on formal definition. In this paper, we develop a fuzzy logic controller for a simulated mobile robot with genetic algorithm and analyze behaviors of the controller from the perspective of the observational emergence. The analysis shows that the fuzzy logic controller has acquired emergent behaviors through the interactions of the underlying fuzzy rules.

1 Introduction

Khepera (see Figure 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. Eight infrared proximity sensors are placed around the robot and each of them embeds an infrared emitter and a receiver.

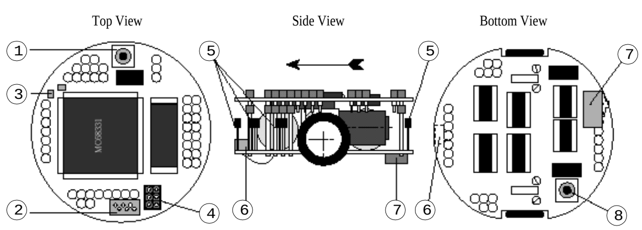


Figure 1: Position of some parts of the robot [1]. ① LEDs. ② Serial line (S) connector. ③ Reset button. ④ Jumpers for the running mode selection. ⑤ Infra-Red proximity sensors. ⑥ Battery recharge connector. ⑦ ON-OFF battery switch. ⑧ Second reset button.

Fuzzy logic controllers (FLC) [2]-[10] have been widely used for behavior-based robots like Khepera 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 determine the optimal parameters. Thus, many researchers have applied evolutionary algorithms to the construction of FLCs in order to automate the procedure of determining the parameters [6]-[9]. Most of them, however, do not provide an in-depth analysis of the behaviors obtained by the FLCs evolved.

Recently, the concept of emergence has been focused as a result of researches in non-linear dynamics, artificial life, complex systems, and behavior-based robotics. Emergence of a system, in a broad way, is said to be the properties or behaviors that cannot easily be predicted from its internal properties. Examples of these phenomena are the flocking behavior in simulated birds from a set of three simple steering behaviors [11], some patterns in the game of life [12], and the "highway" pattern by the artificial Langton ant [13]. In spite of the broadness of emergence phenomena, there is no unified agreement on what emergence is.

One of the first definition on emergence was made by Morgan who defined: "emergence is the denomination of something new which could not be predicted from the elements constituting the preceding condition" or "the product is not a mere sum of the separate elements" [14]. Bedau proposed weak emergence in contrast to strong emergence [15]. According to his definition, properties or behaviors of a system are weakly emergent if and only if they can be derived from the dynamics which governs the time evolution of the system's microstates, only by simulation. Ronald proposed a three-step emergence test [16]. The three steps are design, observation, and surprise. He drew an analogy with the concept of intelligence and the Turing test. Baas [17] defined two emergence: Deducible emergence if there is deductional or computational process or theory, and observational emergence if it cannot be deduced.

In this paper, we construct the FLC whose internal parameters are tuned by GA to control the sensor-based mobile robot, Khepera. In order to do this, we represent the FLC parameters, the fuzzy sets and rules of the FLC, in genetic codes. We show that the evolved behaviors are observationally emergent based on Baas' notion of emergence.

2 Observational Emergence

Emergence has played a major role in the description and discussion of natural and artificial life, and complex systems. In an intuitive point of view, it is used as a name for creation of new structures and properties as the old philosophical statement, "the whole is more than the sum of its parts." For more formal definition of emergence, we have used Baas' notion which will be explained as follows [17].

The definition of emergence starts with a general notion of structures as primitive objects or entities. A structure may be of an abstract or physical nature, e.g., systems, organizations, organisms, machines, concepts, etc. Furthermore, assume that we have some kind of observational mechanism (or family of such) in order to evaluate, observe, and describe the structure. This could be an internal mechanism of the system as well as an external one.

To give a general procedure for how to construct a new structure from a family of old ones, let's start out with a family of structures,

$$\{S_i\}, \quad i \in J \text{ (some index set, finite or infinite).} \tag{1}$$

Then we apply our observational mechanisms, Obs, to obtain properties of the structures,

$$S_i, \quad Obs(S_i). \tag{2}$$

Next we put the S_i's to a family of interactions, Int, using the properties registered under the observation. Hence we get a new kind of structure as follows.

$$S = R(S_i, Obs(S_i), Int), \quad i \in J \tag{3}$$

where R stands for the result of the construction process. Here, S is a second-order structure as opposed to the S_i's which are first-order structures. The interactions may be caused by the structures themselves or imposed by external factors. At each level of construction new properties or new behaviors may emerge, giving room for new interactions, and hence each level is necessary in order to get the last level's properties. Therefore, the Nth order structure is defined as follows.

$$S^N = R(S^{N-1}_{i_{N-1}}, Obs^{N-1}, Int^{N-1}), \quad i_{N-1} \in J_{N-1} \tag{4}$$

where N means the Nth order and i_{N-1} means ith structure of $N-1$th order. From this, we can introduce the definition of emergence as follows.

$$\begin{aligned} &P \text{ is an emergent property of } S^n \iff \\ &\quad P \in Obs^n(S^n), \text{ but } P \notin Obs^n(S^{n-1}_{i_{n-1}}) \text{ for all } i_{n-1} \end{aligned} \tag{5}$$

Emergence is *deducible or computable* if there is a deductional or computational process or theory D such that $P \in Obs^n(S^n)$ can be determined by D from $(S^{n-1}_{i_{n-1}}, Obs^{n-1}, Int^{n-1})$, and observational if P is an emergent property but cannot be deduced as in *deducible*.

3 Evolved Fuzzy Controller

As we use fuzzy logic as the control mechanism of our mobile robot, the first order structures and interactions are determined in terms of fuzzy logic. 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. From the fact that the control in fuzzy logic controller is performed through the interactions (fuzzy inference and defuzzification) of fuzzy rules, we determine the first order structures and interactions as follows.

- First order structures S^1_i = the ith fuzzy rule.
- First order interactions Int^1 = the inference and defuzzification.

3.1 First Order Structures : Fuzzy Rules

To construct the first order structures, fuzzy rules, we first need to define fuzzy sets on both inputs from sensors and outputs to the motors of the mobile robot. Our FLC uses the sensory information of eight proximity sensors as inputs and controls the speed of the two motors on Khepera.

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 terms have the following meanings:

Input		Variable	Output		Variable
VF	:	Very Far	BH	:	Backward High
F	:	Far	B	:	Backward
C	:	Close	F	:	Forward
VC	:	Very Close	FH	:	Forward High

The membership functions of D_i and V_i are all in a triangular form defined by equation (6).

$$\text{triangle}(x, a, b, c) = max\left(min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \tag{6}$$

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 (see Figure 2).

The nth rule can be represented as a fuzzy relation defined by equation (7).

$$\begin{aligned} R_n : &(D^n_0 \times D^n_1 \times D^n_2 \times D^n_3 \times D^n_4 \times D^n_5 \times D^n_6 \times D^n_7) \\ &\to (V^n_0, V^n_1) \end{aligned} \tag{7}$$

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

$$\begin{aligned} &\mu_{RV^n_0}(d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, v_0) \\ &\qquad = f(\mu_{D^n_0}(d_0), \cdots, \mu_{D^n_7}(d_7), \mu_{V^n_0}(v_0)) \end{aligned} \tag{8}$$

$$\begin{aligned} &\mu_{RV^n_1}(d_0, d_1, d_2, d_3, d_4, d_5, d_6, d_7, v_1) \\ &\qquad = f(\mu_{D^n_0}(d_0), \cdots, \mu_{D^n_7}(d_7), \mu_{V^n_1}(v_1)) \end{aligned} \tag{9}$$

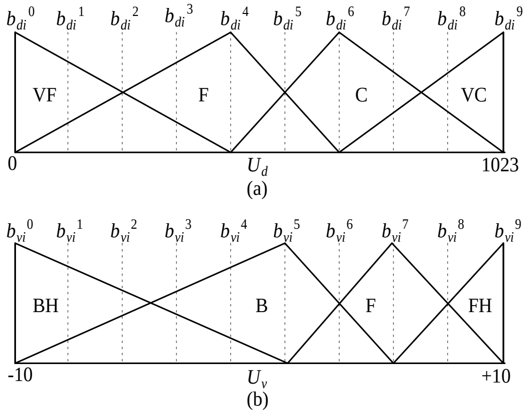


Figure 2: The membership functions of input (a) and output (b) variables.

3.2 Interactions

Here, we define the interactions of the first order structures (fuzzy rules). The fuzzy rules can interact with each other by fuzzy reasoning and defuzzifiation.

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

$$\begin{aligned} V_0' &= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} (D_0^n \times \cdots \times D_7^n \rightarrow V_0^n) \\ &= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} RV_0^n \end{aligned} \tag{10}$$

$$\begin{aligned} V_1' &= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} (D_0^n \times \cdots \times D_7^n \rightarrow V_1^n) \\ &= (D_0^{n'} \times \cdots \times D_7^{n'}) \circ \bigcup_{n=0}^{N-1} RV_1^n \end{aligned} \tag{11}$$

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

$$\begin{aligned} \mu_{V_0'} &= \bigcup_{n=0}^{N-1} \underbrace{\left\{ \vee_{d_0} \left[\mu_{D_0^{n'}}(d_0) \wedge \mu_{D_0^n}(d_0) \right] \right\}}_{\omega_0} \wedge \cdots \wedge \\ & \quad \underbrace{\left\{ \vee_{d_7} \left[\mu_{D_7^{n'}}(d_7) \wedge \mu_{D_7^n}(d_7) \right] \right\}}_{\omega_7} \wedge \mu_{V_0^n}(v_0) \\ &= \bigcup_{n=0}^{N-1} \underbrace{(\omega_0 \wedge \cdots \wedge \omega_7)}_{\text{firing strength}} \wedge \mu_{V_0^n}(v_0) \end{aligned} \tag{12}$$

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

$$\mu_{V_1'} = \bigcup_{n=0}^{N-1} \underbrace{(\omega_0 \wedge \cdots \wedge \omega_7)}_{\text{firing strength}} \wedge \mu_{V_1^n}(v_1) \tag{13}$$

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

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]-[21], the center of gravity method is used because it is widely used and also appropriate for our system to control the mobile robot.

$$\bar{v}_0 = \frac{\int_{v_0} \mu_{v_0}(v_0) v_0 dv_0}{\int_{v_0} \mu_{v_0}(v_0) dv_0} \tag{14}$$

$$\bar{v}_1 = \frac{\int_{v_1} \mu_{v_1}(v_1) v_1 dv_1}{\int_{v_1} \mu_{v_1}(v_1) dv_1} \tag{15}$$

3.3 Evolution

A genetic algorithm (GA) is a search technique based on the mechanics of natural selection and natural genetics [22]. 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.

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.

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,

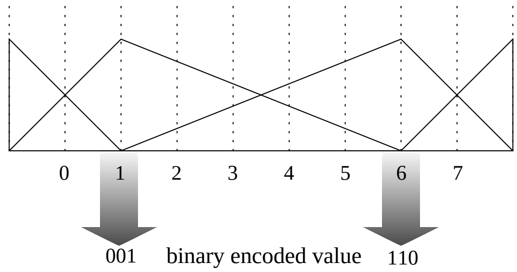


Figure 3: Encoding of membership function.

and maximally ten rules are encoded. 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 and only six bits are required to encode two fuzzy sets (see Figure 3).

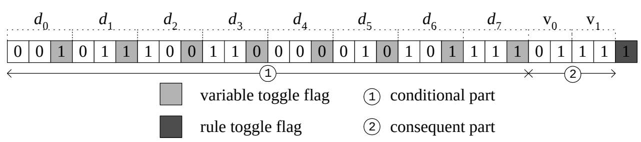


Figure 4: Encoding of a rule.

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 Figure 4) 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 Figure 4 designates whether this rule participates in fuzzy inference process or not. Therefore, Figure 4 can be decoded as follows:

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

To use GA for the evolution of individuals representing FLC for the mobile robot, a fitness function is defined by equation (16).

$$\begin{aligned} f = &\alpha(\text{number of collisions}) + \beta(\text{moving distance}) + \\ &\gamma(\text{number of rules}) + \delta(\text{number of fuzzy sets}) + \\ &\epsilon(\text{check points}) \end{aligned} \tag{16}$$

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.

4 Emergence Analysis

Experiments have been performed on a 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 Figure 5. Each individual is evaluated for five thousands times of sensor sampling. In early generations, the

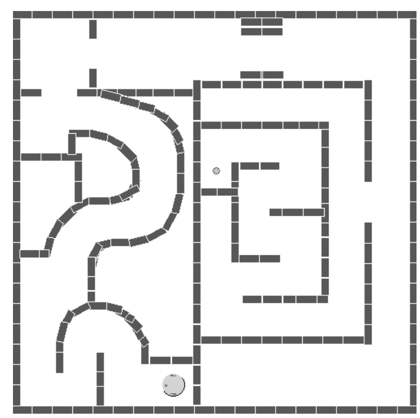


Figure 5: The environment used for evolving the robot.

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 generation passes. Although we have used an elite preserving strategy in the selection process [22], 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 fuzzy rules of the best individual are shown in Table 1. For the demonstration of observational emergence, we once again state the parameter settings of emergence.

- First order structures $S_i^1 =$ the ith fuzzy rule.
- First order interactions $Int^1 =$ the fuzzy inference and defuzzification.
- $Obs^1 = Obs^2 =$ the behavioral properties of structure.
- Second order structure $S^2 = R(S_{i_1}^1, Obs^1, Int^1)$ = the evolved fuzzy logic controller.

4.1 Turning Around

Turning around from a dead end is an important behavior that the robot should acquire. From the simulation, we find that

Table 1: Evolved fuzzy rules

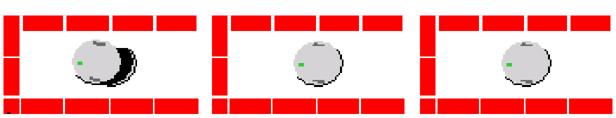
Rule 1	IF (d_2 =C) and (d_5 =VF) and (d_7 =VC) THEN (v_0 =BH) and (v_1 =B)
Rule 2	IF (d_4 =VF) THEN (v_0 =FH) and (v_1 =F)
Rule 3	IF (d_1 =VC) and (d_2 =F) and (d_4 =C) and (d_7 =VC) THEN (v_0 =BH) and (v_1 =B)
Rule 4	IF (d_2 =F) and (d_3 =F) and (d_6 =VC) THEN (v_0 =F) and (v_1 =FH)
Rule 5	IF (d_4 =VC) 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)

the three first order strcutures $S^1_{2_1}$, $S^1_{5_1}$, and $S^1_{7_1}$ in Table 2 are interacting in turning around situation.

Table 2: $S^1_{2_1}$, $S^1_{5_1}$, and $S^1_{7_1}$

$S^1_{2_1}$(=Rule 2) : IF	(d_4 =VF) THEN (v_0 =FH) and (v_1 =F)
$S^1_{5_1}$(=Rule 5) : IF	(d_4 =VC) THEN (v_0 =BH) and (v_1 =F)
$S^1_{7_1}$(=Rule 7) : IF	(d_0 =VF) and (d_4 =F) and (d_5 =C) THEN (v_0 =BH) and (v_1 =F)

Figure 6 shows the Obs's of the three structures. As you can see, $Obs(S^1_{2_1})$ shows that the robot stops after some moves from the start position (see also Figure 7). $Obs(S^1_{5_1})$ and $Obs(S^1_{7_1})$ show that the robot does not move at all. On the other hand, Figure 8 shows Obs^2 of the fuzzy controller composed of three structures, $S^1_{2_1}$, $S^1_{5_1}$, and $S^1_{7_1}$ interacting with each other. During the steps from 1 to 61 in Figure 8, the robot uses all the three first order structures. Figure 9

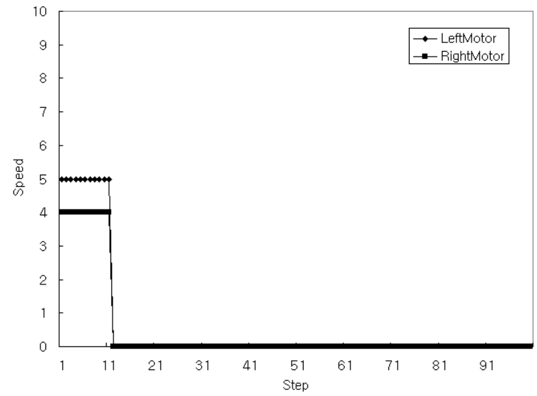


(a) $Obs(S^1_{2_1})$ (b) $Obs(S^1_{5_1})$ (c) $Obs(S^1_{7_1})$

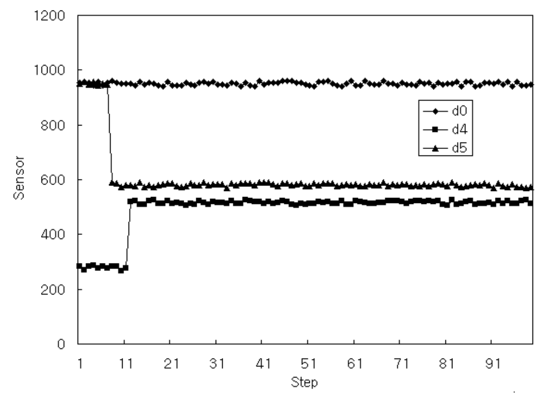
Figure 6: Obs of three structures from step 1 to 61.

shows another view of Obs^2 when the robot turns around: Figure 9 (a) shows the sensing values of the related sensors from step 1 to step 100. Figure 9 (b) shows the activation levels of related rules, 2, 5, and 7, and Figure 9 (c) shows the speed changes of the two motors during this process.

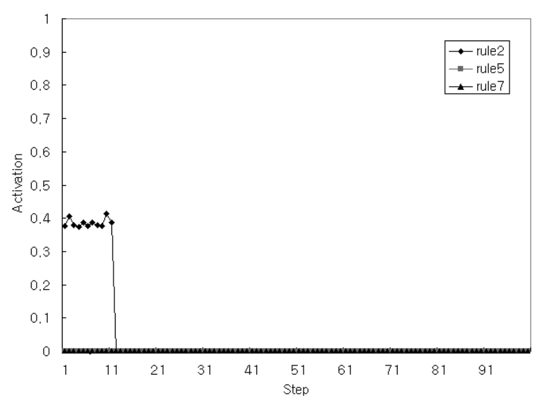
As can be seen in Figure 9, the interactions of the three



(a) Speed according to the activation of rule 2

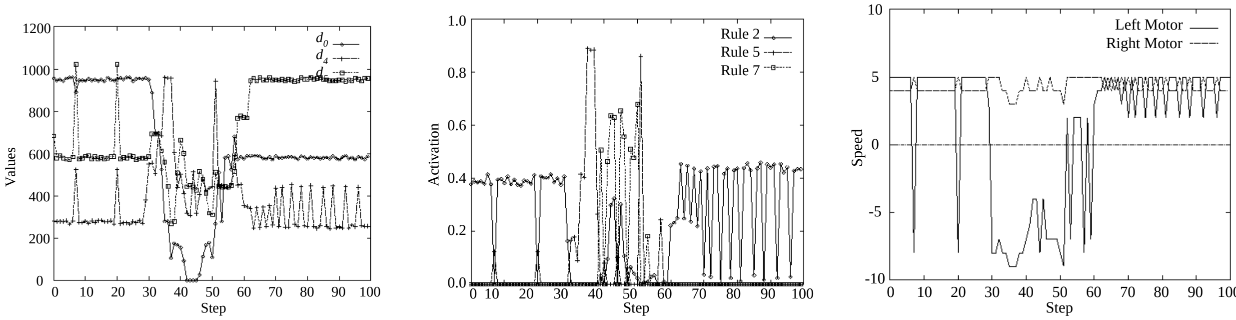


(b) d_0, d_4, and d_5 sensor values



(c) Activation of rule 2

Figure 7: Analysis of $Obs^1_{2_1}$.

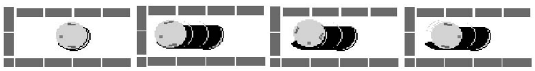


(a) d_0, d_4, and d_5 sensor values

(b) Activation of related rules

(c) Stepwise speed change of the robot

Figure 9: Analysis of Obs^2.



(a) step 1

(b) step 31

(c) step 41

(d) step 61

Figure 8: Obs^2.

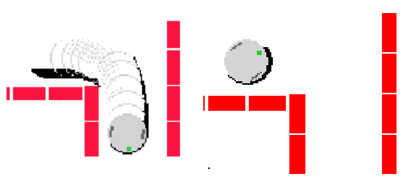
first order structures, $S^1_{2_1}$, $S^1_{5_1}$, and $S^1_{7_1}$, make the $Obs^2(S^2)$ different from the $Obs^1(S^1_{i_1})$ of first order. This implies that

$$P \in Obs^2(S^2), \text{but } P \notin Obs^1(S^1_{i_1}) \text{ for all } i_1 \quad \text{with } P = \text{“Turn Around”} \tag{17}$$

Therefore, we can conclude that the $Obs^2(S^2)$ is the emergent behavior by the definition in equation (5).

4.2 Smooth Cornering

At a corner, the robot should turn the corner to the left or right as safely and smoothly as possible. From the simulation, we find that the two first order structures $S^1_{2_1}$ and $S^1_{7_1}$ shown in Table 2 are interacting in conrer situations. Figure 10 shows the two structures, $Obs(S^1_{2_1})$ and $Obs(S^1_{7_1})$. As you can



(a) $Obs(S^1_{2_1})$

(b) $Obs(S^1_{7_1})$

Figure 10: Obs of three structures from step 1 to 90.

see, $Obs(S^1_{2_1})$ shows that the robot turns the corner with difficulty, and also there are some bumps at the corner (see also Figure 11). $Obs(S^1_{7_1})$ shows that the robot rather moves backward. On the other hand, Figure 12 shows Obs^2 of the fuzzy controller composed of two structures $S^1_{2_1}$ and $S^1_{7_1}$ interacting with each other. During the steps from 1 to 90 in Figure 12, the robot uses all the two first order structures. Figure 13 shows another view of Obs^2 when the robot turns the corner: Figure 13 (a) shows the activation levels of related rules, 2 and 7, and Figure 13 (b) shows the speed changes of the two motors during this process.

As can be seen in Figure 13, the interactions of the two first order structures, $S^1_{2_1}$ and $S^1_{7_1}$, make the $Obs^2(S^2)$ different from the $Obs^1(S^1_{i_1})$ of first order. This implies that

$$P \in Obs^2(S^2), \text{but } P \notin Obs^n(S^1_{i_1}) \text{ for all } i_1 \quad \text{with } P = \text{“Smooth Cornering”} \tag{18}$$

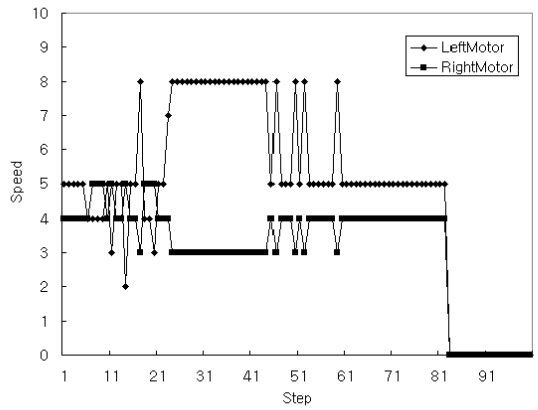
Therefore, we can conclude that the $Obs^2(S^2)$ is the emergent behavior by the definition in equation (5).

5 Conclusion

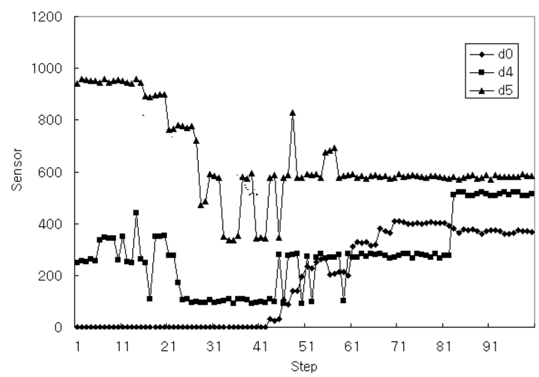
In this paper, we have analyzed the FLC of a mobile robot evolved by a GA with Baas' definition of emergence. The FLC finally constructed by evolution consists of just seven rules with one to three input variables and even more only three of them are used to get to the goal position, although, theoretically, there can be 2^{28} rules because 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.

The robot has obtained proper rules for several behaviors like smooth cornering and turning around to navigate in the complex environment. These rules form what Baas calls first order structures. We have shown that these first structures and their interactions give rise to the emergent behaviors or observational emergence of second order structures.

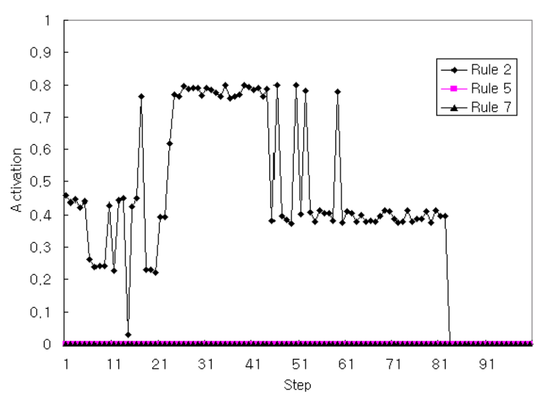
As further research, the third or higher order structures are to be researched based on the second order structures shown as the robot's behaviors in this paper. This can be researched in several ways. One of them is to study the group behaviors of multiple robot agents each of which consists of second order structures.



(a) Speed according to the activation of rule 2

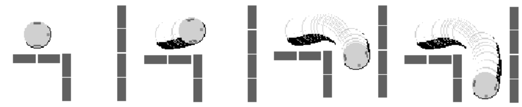


(b) d_0, d_4, and d_5 sensor values



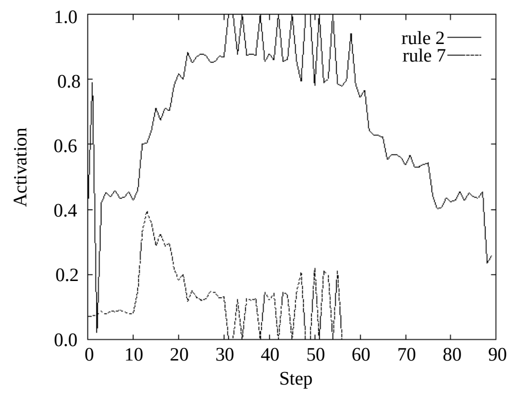
(c) Activation of rule 2

Figure 11: Analysis of $Obs^1_{2_1}$.

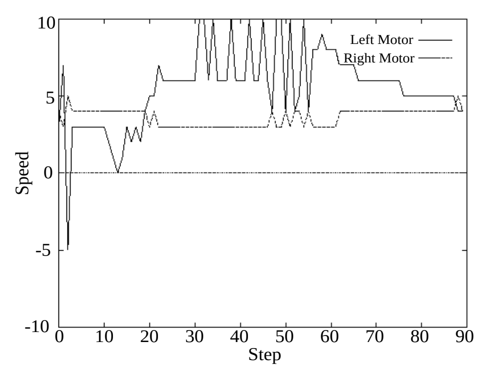


(a) step 1 (b) step 28 (c) step 68 (d) step 90

Figure 12: Cornering.



(a) Activation of related rules



(b) Stepwise speed change of the robot

Figure 13: Analysis of Obs^2.

Bibliography

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

[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] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] C.W. Reynolds, “Flocks, herds, and schools: A distributed behavioral model,” *Computer Graphics*, vol. 21, pp. 25-34, 1987.

[12] E.R. Berlekamp, J.H. Conway and R.K. Guy, “Winning ways for your mathematical plays,” *Academic Press*, vol. 2, pp. 817-850, 1982.

[13] I. Stewart, “The ultimate in anty-particles,” *Scientific American*, vol. 271, pp. 104-107, 1994.

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

[15] M.A. Bedau, “Weak emergence,” *Philosophical Perspectives: Mind, Causation, and World*, vol. 11, pp. 375-399, 1997.

[16] E.M.A. Ronald, M. Sipper and M.S. Capcarrere, “Design, observation, surprise! A test of emergence,” *Artificial Life*, vol. 5, pp. 225-239, 1999.

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

[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] 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.

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