Control and Cybernetics

vol. 27 (1998) No. 4

Behavior analysis of genetic fuzzy controller for an autonomous robot[1]

by

Sung-Bae Cho and Seung-Ik Lee

Dept. of Computer Science, Yonsei University,
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749,
Korea

Abstract: To program an autonomous robot so that it acts reliably in a dynamic environment is a very hard task. Towards a promising approach to this problem, we have developed a genetic fuzzy controller for a mobile robot, and showed the possibility by applying it to a simulated robot called Khepera. The robot gets input from eight infrared sensors and operates two motors according to the fuzzy inference based on the sensory input. This paper attempts to analyze the adaptive behaviors of the controller by using automata, which indicates the emergence of several strategies to make the robot to navigate the complex space without bumping agains the walls and obstacles.

Keywords: softcomputing, genetic fuzzy controller, mobile robot, automata, behavior analysis

1. Introduction

It is quite difficult to construct the optimal controller that appropriately adapts to the ever-changing environments. This is because of missing necessary information at design stage, the unpredictability of the environment dynamics, and the inherent noise of the sensors and actuators (Dorigo, 1996). Clearly, an autonomous robot that can acquire knowledge by interaction with the environment and subsequently adapt and change its behavior in the run time could greatly simplify the work of its designer. As a promising approach to the learning of autonomous robot, behavior-based robotics has recently appeared (Brooks, 1986; Dorigo and Schnepf, 1993).

One of the key points of this approach is not to give the robot information about the environment but to let the robot find the knowledge by itself. With

[1]This work has been supported in part by a grant no. SC-13 from the Ministry of Science and Technology in Korea.

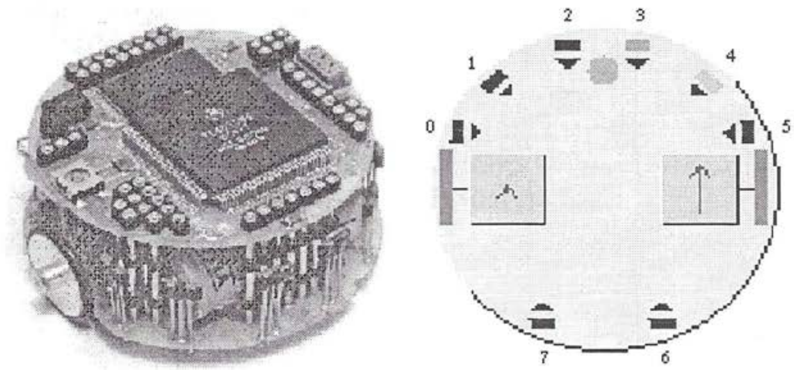


Figure 1. Khepera robot and the simulated representation.

this approach, a number of researchers have successfully employed evolutionary procedures to develop the control system of simulated robots (Beer and Gallagher, 1992; Cliff, Harvey and Husbands, 1993; Parisi, Cecconi and Nolfi, 1990). The rich variety of structures, which have appeared during evolution and the large number of evolved behaviors, have empirically demonstrated the power and generality of the evolutionary algorithms. However, this approach suffers from the difficulty of analyzing the control system evolved, which prohibits the designer from making use of some domain knowledge to design the control system by an evolutionary approach.

To work out this problem, we proposed a fuzzy system for a behavior-based robot, and presented an evolutionary approach to determine the structure and parameters of the fuzzy controller (Cho and Lee, 1998). In this paper, we attempt to analyze the genetic fuzzy controller developed to control the simulated robot called Khepera. We also show that the adaptive behaviors result from the interaction of several primitive low-level strategies acquired through the evolutionary process.

2. Autonomous robot

For the simulation, we have used the Khepera robot that is circular, compact and robust as shown in Figure 1. This is a miniature robot that has diameter of $55mm$, height of $30mm$, and weight of $70g$. The robot is supported by two wheels and two small Teflon balls placed under its platform. The wheels are controlled by two DC motors with an incremental encoder (12 pulses per mm of robot advancement) and can rotate in both directions. The geometrical shape and the motor layout of Khepera make the robot to navigate in sophisticated environment even when its control system is immature.

It is provided with eight infrared proximity sensors placed around its body which are based on emission and reception of infrared light. Each receptor can measure both the ambient infrared light and the reflected infrared light emitted by the robot itself. Several new single sensors and complete modules, such as a stereo-vision module and a gripper module, can be easily added, due to the hardware and software modularity of the system.

Dedicated to Khepera, the simulated mobile robot (Michel, 1995) includes eight infrared sensors (small rectangles) allowing it to detect by reflection the proximity of objects in front of it, behind it, and to the right and left sides of it. Each sensor returns a value ranging between 0 and 1023 represented in gradual color levels. 0 means that no object is perceived whereas 1023 means that an object is very close to the sensor (almost touching the sensor). Intermediate values may give an approximate idea of the distance between the sensor and the object. Each motor can take a speed value ranging between -10 and $+10$. The size of arrows on the motors in Figure 1 indicates the amount of speed.

3. Genetic fuzzy controller

In order to operate the robot introduced at the previous section, we have developed a fuzzy controller of which the internal parameters are adapted with genetic algorithms. A fuzzy inference system provides a computing framework based on the concepts of fuzzy sets, fuzzy if-then rules, and fuzzy reasoning. The basic structure consists of a fuzzy rulebase, reasoning mechanism, and defuzzification mechanism. A fuzzy rulebase is a set of fuzzy rules that are expressed as follows:

Rule 1: If (x_1 is A_1^1) and (x_2 is A_2^1) and ... and (x_n is A_n^1), then y is B^1
Rule 2: If (x_1 is A_1^2) and (x_2 is A_2^2) and ... and (x_n is A_n^2), then y is B^2
...
Rule m: If (x_1 is A_1^m) and (x_2 is A_2^m) and ... and (x_n is A_n^m), then y is B^m

where x_j $(1 \leq j \leq n)$ are input variables, y is output variable, and A_j^i and B^i $(1 \leq i \leq m)$ are fuzzy sets which are characterized by membership functions. In our simulation, the numbers of input and output variables are eight and two, respectively.

In order to facilitate the design of the controller, we adopt the following four fuzzy sets for the input and output parameters:

Input : 8 values from infrared sensors ($0 \sim 1023$)
Fuzzy set : I = {VF, F, C, VC}
VF (Very Far)
F (Far)
C (Close)
VC (Very Close)

Output : 2 values from motors ($-10 \sim +10$)
Fuzzy set : O = {BH, B, F, FH}
BH (Backward High)
B (Backward)
F (Forward)
FH (Forward High)

The fuzzy sets could be simplified if we used partition of less number of fuzzy values, but the four values make the robot navigate smoothly. Even though the fuzzy sets consist of four fuzzy values, the exact partitioning of input/output spaces depends on membership functions which are determined by genetic algorithms in our approach. Triangular shapes specify the membership function. A parameter value divides the range ($0 \sim 1023$ for input and $-10 \sim +10$ for output) by ten equidistance segments.

For fuzzy inference we use correlation minimum method, which truncates the consequent fuzzy region at the truth of the premise (Kosko, 1992). The firing strength, μ_i, of the ith rule is defined as follows.

$$\mu_i = \min(I_{i0}(x_0), I_{i1}(x_1), \cdots, I_{ij}(x_j)), \tag{1}$$

where j is the number of input variables, and I_{ij} is the fuzzy membership function defined at the jth input variable of the ith rule.

Finally, centroid defuzzification method is adopted to yield the expected value, y_l^*, of the solution fuzzy region, as follows.

$$y_l^* = \frac{\sum_{i=0}^{m} \mu_i \overline{y_i}}{\sum_{i=0}^{m} \mu_i}, \tag{2}$$

where $\overline{y_i}$ is the ith domain value. Figure 2 shows an example of the fuzzy inference and defuzzification used in this paper. In this figure, rules 5 and 7 are activated and produce two output values: 3 and 4.

In order to robustly determine the shape and number of membership functions in fuzzy rules, genetic algorithm has been utilized. This approach reduces the burden of human operators to decide the structure of fuzzy rules. Genetic algorithm (GA) is considered as an effective method for optimization (Goldberg, 1989), and several hybrid methods with fuzzy logic have been recently proposed. Figure 3 shows the overall diagram of the proposed system.

The parameters in the fuzzy system are represented as a gene, and the performance with the Khepera simulator decides whether it can produce offsprings

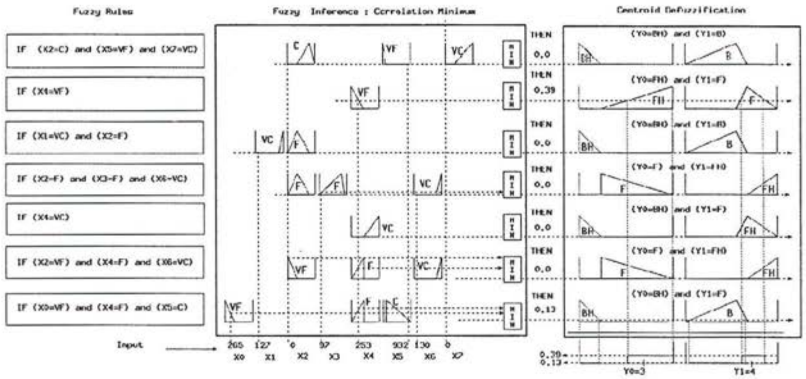


Figure 2. An example of the fuzzy inference and defuzzification.

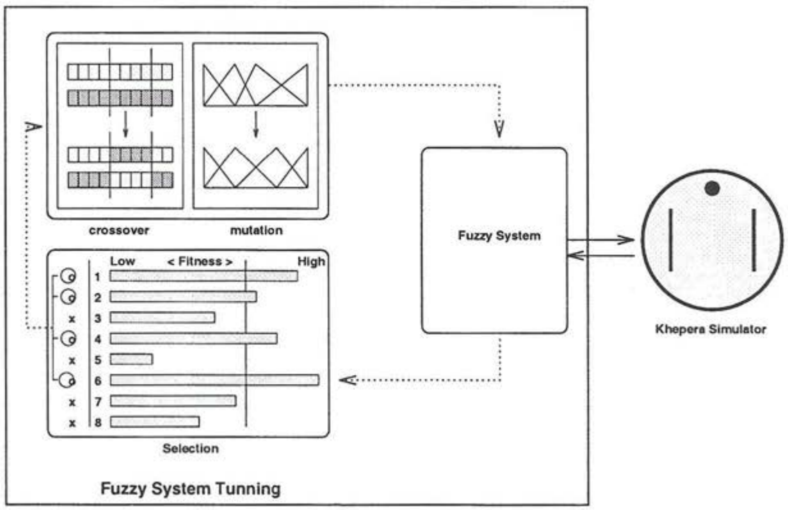


Figure 3. Schematic diagram of the genetic fuzzy system.

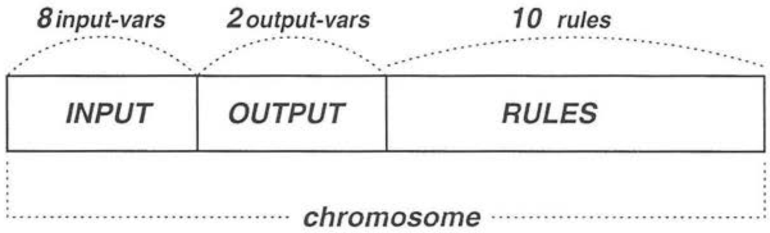


Figure 4. Gene code for encoding the fuzzy system.

with the genetic operators. In the figure, four genes of numbers 1, 2, 4 and 6 are selected as candidates for the next generation, and the crossover is applied to them.

To get a success in the application of genetic algorithm, it is quite important to devise a gene coding scheme appropriate to the problem. In this paper, we should incorporate the input and output membership functions and the rules as a gene code as shown in Figure 4 which encodes the eight input parameters, two output parameters and maximum 10 rules. For details on the gene encoding scheme, see the recent publication (Cho and Lee, 1998).

Another important issue in the genetic algorithm is to determine a proper fitness measure for the problem. In this paper we make the fitness function decrease as the robot collides with the walls, and increase as it moves farther from the start point. In addition, a couple of factors are included to induce the compact fuzzy system by preferring to the smaller number of rules and membership functions. The fitness function is as follows.

$$\begin{aligned} \text{fitness} &= \alpha \times \text{no. of collisions} \\ &+ \beta \times \text{distance moved} \\ &+ \gamma \times \text{no. of rules} \\ &+ \delta \times \text{no. of membership functions} \\ &+ \epsilon \times \text{no. of check points reached,} \end{aligned} \tag{3}$$

where $\alpha = -3$, $\beta = 1$, $\gamma = -100$, $\delta = -10$, and $\epsilon = 500$.

The coefficients might be determined by another optimization technique, but in this paper we have just selected them empirically. The fitness would increase as the robot goes farther from the start point while passing by more check points. The fitness would decrease as the robot collides with the walls or the numbers of rules and membership functions get larger. In order to expedite the evolution, we put several check points along with the pathways which will be removed later.

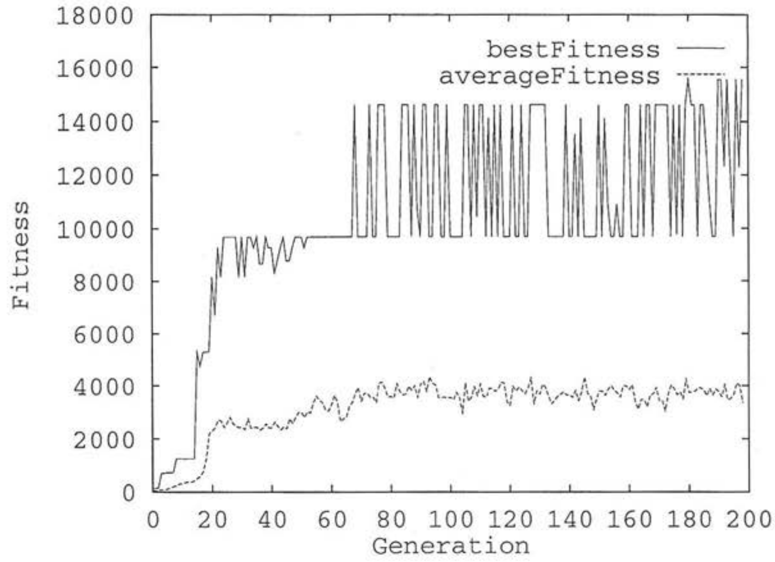


Figure 5. Fitness change.

4. Simulation

The Khepera simulator has been written in C++ (Michel, 1995), and the simulation was conducted in SUN Sparc 10 workstation. We initialized 200 chromosomes at random, each of which was developed to a fuzzy controller for the robot. Each robot operates within 5000 sensor sampling time, and produces the performance value according to the fitness function.

Figure 5 shows the best and average fitness changes in the course of simulation. As the figure depicts, the performance increases as the generation goes, and a robot navigated successfully at the given environment has been obtained at less than 100 generations. It can be seen that the fitness is radically increased at the beginning stage, but there is nearly no change after 90 generations except some oscillation. This has occurred mainly because the elite preserving strategy has not been incorporated in producing the next generation. The fitness jumps to a high value when the robot can escape from the closed corridor. Around the 67*th* generation the best individuals already perform a near optimal behavior. They navigate smoothly not to bump into walls and corners, and maintain a straight trajectory whenever it is possible.

Figure 6 shows the trajectories that the robot has made during the simulation. These results are highly reliable and have been replicated in many runs of the experiment. In the beginning of the evolution the individuals evolved a frontal direction of motion, corresponding to the side where more sensors are available. Those individuals that moved in the other direction got stuck in a

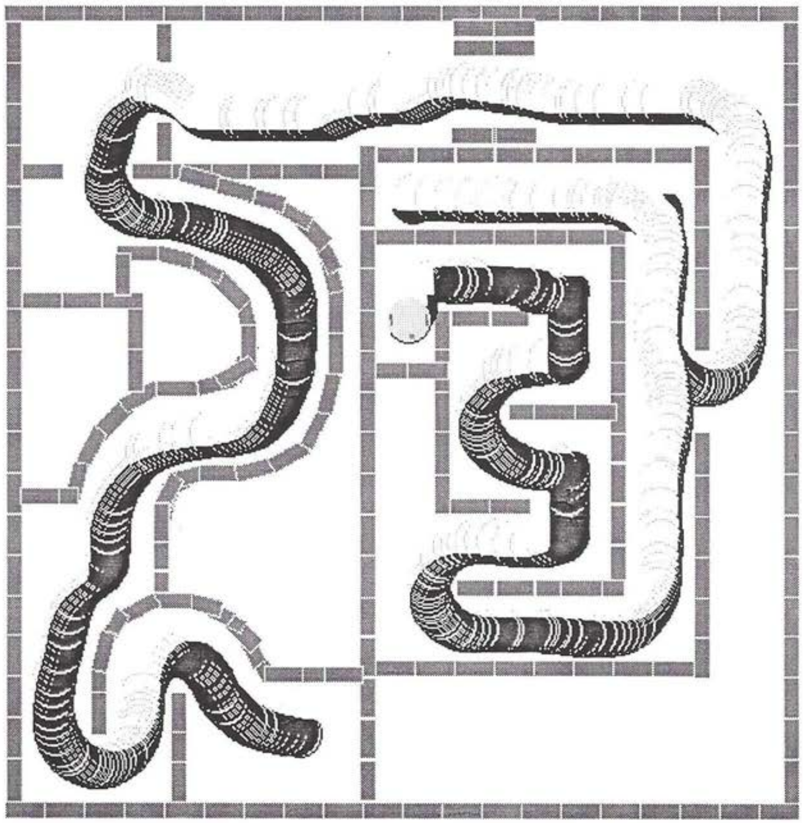


Figure 6. Trajectories of the robot.

corner without being able to detect it, and soon disappeared from the population. The controller for this robot consists of only seven effective rules, which are generated through the evolutionary process as follows.

$$\begin{array}{ll}
\text{Rule 1: } & \text{If } (x_2 = C) \text{ and } (x_5 = VF) \text{ and } (x_7 = VC) \\
 & \quad \text{Then } (y_0 = BH) \text{ and } (y_1 = B) \\
\text{Rule 2: } & \text{If } (x_4 = VF) \\
 & \quad \text{Then } (y_0 = FH) \text{ and } (y_1 = F) \\
\text{Rule 3: } & \text{If } (x_1 = VC) \text{ and } (x_2 = F) \text{ and} \\
 & \quad\ (x_4 = C) \text{ and } (x_7 = VC) \\
 & \quad \text{Then } (y_0 = BH) \text{ and } (y_1 = B) \\
\text{Rule 4: } & \text{If } (x_2 = F) \text{ and } (x_3 = F) \text{ and } (x_6 = VC) \\
 & \quad \text{Then } (y_0 = F) \text{ and } (y_1 = FH) \\
\text{Rule 5: } & \text{If } (x_4 = VC) \\
 & \quad \text{Then } (y_0 = BH) \text{ and } (y_1 = F) \\
\text{Rule 6: } & \text{If } (x_2 = VF) \text{ and } (x_4 = F) \text{ and } (x_6 = VC) \\
 & \quad \text{Then } (y_0 = F) \text{ and } (y_1 = FH) \\
\text{Rule 7: } & \text{If } (x_0 = VF) \text{ and } (x_4 = F) \text{ and } (x_5 = C) \\
 & \quad \text{Then } (y_0 = BH) \text{ and } (y_1 = F)
\end{array}$$

Even though we did not give any hints to the system, several effective rules to control the mobile robot appropriately at a number of different cases have emerged through the evolution. The overall behavioral model can be depicted as Figure 7.

The rule 2 triggers the state of "Obstacle Avoidance," the rules 2 and 7 cooperatively induce the state of "Wall Following," and the rule 5 activates the state of "Impact Avoidance." This result dictates that the evolutionary approach is quite useful to design a flexible and efficient fuzzy systems to control mobile robot.

For instance, Figure 8 shows the snapshots of the robot that escapes from the closed corridor. When the robot arrives at the closed corridor the internal state of the robot changes to "Impact Avoidance" which is governed by rule 5, while the usual "Wall Following" state is activated by rules 2 and 7. Figure 9(a) depicts the speed of the two motors with respect to the activation levels of rule 5. As can be seen from this figure, the robot turns left as soon as the rule 5 is activated. Figure 9(b), (c) and (d) show the changes of the sensor values, the activation levels of the rules, and the speed of left and right motors, respectively.

5. Concluding remarks

Building a genetic fuzzy controller is by all means a consistent approach to the problem of automatically adapting the behavior of a mobile robot in a changing environment. This paper has presented a fuzzy system to control the Khepera robot, and utilized genetic algorithm to optimize the internal parameters in the system. A successful controller generated consists of only seven effective

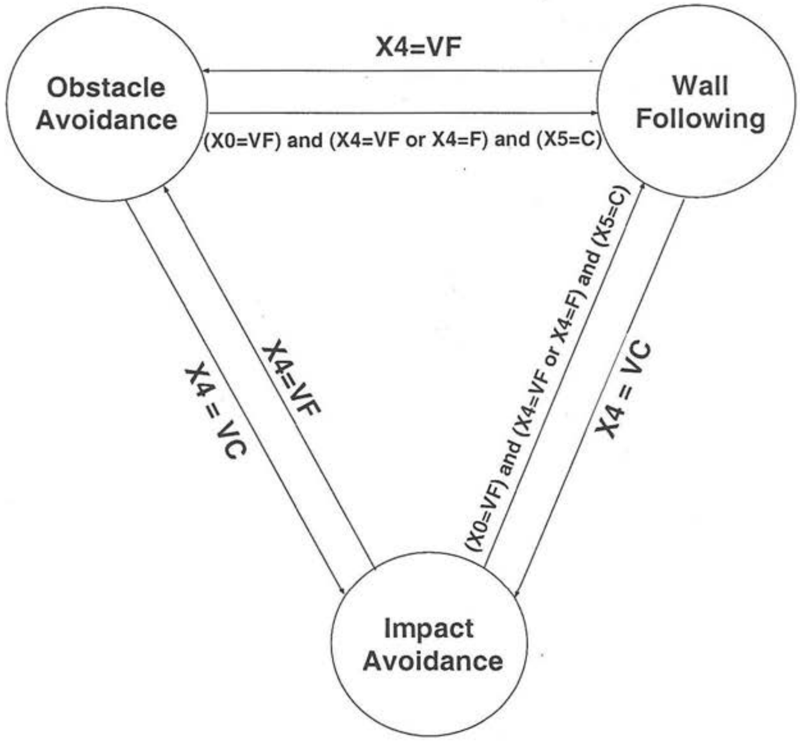


Figure 7. Behavior model for the robot evolved.

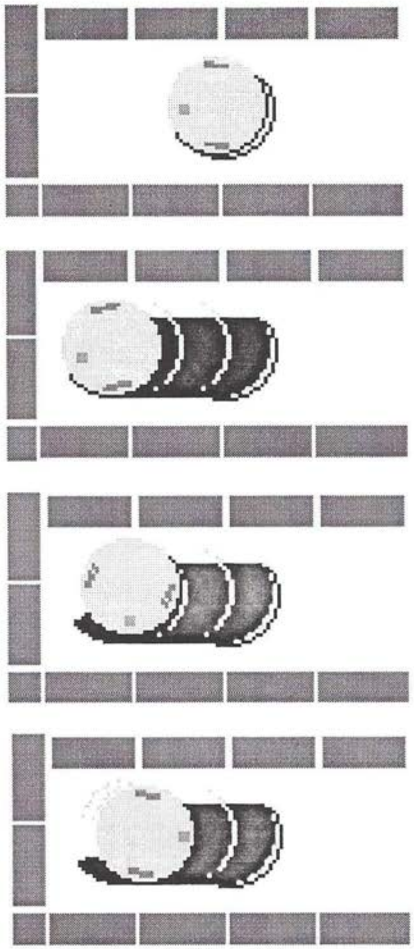
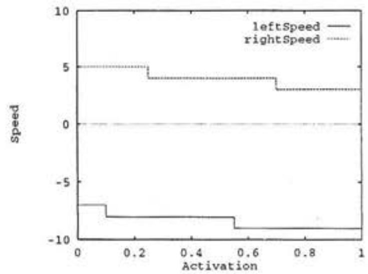
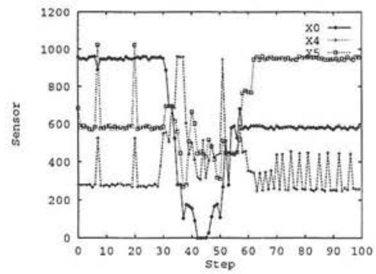


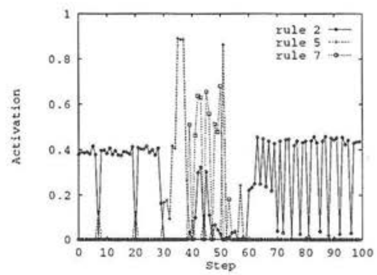
Figure 8. Snapshots of the robot escaping from the closed corridor.



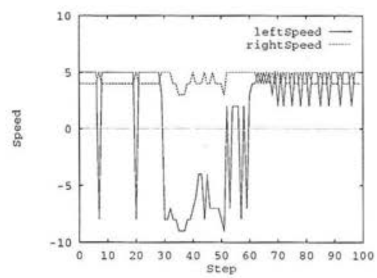
(a) speed of two motors with respect to the activation of rule 5



(b) changes of the sensor values



(c) activation levels of the rules



(d) speed of left and right motors

Figure 9. Internal states of the robot in the closed corridor.

rules, which shows the evolution finds out the optimal set of rules to control the robot. The simulation result shows that the evolutionary approach is quite useful to design a flexible and efficient fuzzy systems to control mobile robot. Nevertheless, further efforts are required to deal with the stability issue of the proposed controller.

References

BEER, R.D. and GALLAGHER, J.C. (1992) Evolving dynamical neural networks for adaptive behavior. *Adapt. Beh.*, **1**, 91–122.

BROOKS, R.A. (1986) A robust layered control system for a mobile robot. *IEEE Trans. Robotics and Automation*, **2**, 1, 14–23.

CLIFF, D., HARVEY, I. and HUSBANDS, P. (1993) Explorations in evolutionary Robotics. *Adapt. Beh.*, **2**, 73–110.

CHO, S.-B. and LEE, S.-I. (1998) Mobile robot learning by evolution of fuzzy controller. *Int. Journal of Intelligent and Fuzzy Systems*, **6**, 1, 91–97.

DORIGO, M. and SCHNEPF, U. (1993) Genetic-based machine learning and behavior based robotics: A new synthesis. *IEEE Trans. Syst. Man, Cybern.*, **23**, 141–154.

DORIGO, M. (1996) Introduction to the special issue on learning autonomous robots. *IEEE Trans. Syst. Man, Cybern.*, **26**, 361–364.

GOLDBERG, D.E. (1989) *Genetic Algorithms in Search Optimization & Machine Learning*. Addison-Wesley.

KOSKO, B. (1992) *Neural Networks and Fuzzy Systems*. Prentice-Hall.

MICHEL, O. (1995) *Khepera Simulator Version 1.0 User Manual*.

PARISI, D., CECCONI, F. and NOLFI, S. (1990) Econets: Neural networks that learn in an environment. *Network*, **1**, 149–168.