Control and Cybernetics

vol. **25** (1996) No. 2

Finding control knowledge for multiple mobile robots using fuzzy classifier systems

by

Ken Nakaoka*, Takeshi Furuhashi*, Hiroshi Maeda, Yoshiki Uchikawa***

*Department of Information Electronics, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, 464-01, Japan
E-mail: furuhashi@nuee.nagoya-u.ac.jp

**Suzuka University of Medical Science and Technology
1000-1, Kishioka-cho, Suzuka, 510-02 Japan

Abstract: This paper details our attempts to find control knowledge using Fuzzy Classifier Systems (FCSs). Three mobile robots are equipped with the FCS controllers and each robot with an FCS moves avoiding collision with other robots. This paper also presents new payoffs and credits. The new methods make it possible to fully utilize a powerful feature of the genetic algorithm, namely the effectiveness of crossover operations. Simulations are done to show that each FCS can find fuzzy rules for collision avoidance in a complex, changing environment.

Keywords: fuzzy classifier system, genetic algorithm, collision avoidance.

1. Introduction

Fuzzy controls, described in linguistic IF-THEN rules, have been widely used in industry in view of their high degree of performance in human-computer interactions. Demand for fuzzy inference systems which can describe complex, multi-input/output situations is growing. In application of fuzzy inference to large scale systems, there are the following difficulties: 1) The number of fuzzy rules becomes extremely large; 2)It takes a great deal of time to identify the objects; and, 3) It is difficult to obtain the complete input/output data of the systems.

The authors have studied applications of the stimulus-response type Fuzzy Classifier Systems (FCS), Valenzuela-Rendon (1991), to knowledge-finding in large scale systems, Furuhashi, Nakaoka, Morikawa, Uchikawa (1993;1994). The

FCS is a Classifier System (CS), Goldberg (1989), into which a fuzzy rule base and fuzzy inference are introduced. The FCS can obtain appropriate fuzzy rules using reinforcement learning based on payoffs. A feature of the FCS is that the designer of the inference system has only to judge the final performance of the system: the FCS finds detailed fuzzy rules according to the designer's judgment.

Research on Q-learning and Profit Sharing for reinforcement learning has been performed by Nagayoshi (1994), Iwashita, Yamamura, Kobayashi (1994). In these references the space where the robot moves around is divided into small subspaces, and the information of the obstacles and the goal, as well as the behavior of the robot are handled on the basis of crisply divided subspaces. Learning is done under discrete input/output relations. On the contrary, the FCS can handle continuous variables and is expected to learn in a more complex, changing environment. However, the control problem studied in Furuhashi, Nakaoka, Morikawa, Uchikawa (1993;1994), was very simple, merely steering a ship to reach the goal.

This paper details our attempts to find control knowledge for use by multiple mobile robots avoiding collision with each other. Three mobile robots are equipped with FCS controllers. Each robot moves according to its own fuzzy rules, which are found by each FCS in the complex, varying environment. This paper also presents new payoffs and credits, i.e. (1) payoffs and credits based on the performance of the system, and (2) payoffs and credits based on the number of valid membership functions. The proposed payoffs and credits are effective for finding valid rules from a large number of possibilities, e.g. 62,500 in this paper.

2. Problem formulation

We have attempted to find control knowledge using three mobile robots equipped with FCS controllers. The task presented is to steer each robot to reach a goal avoiding collision with other robots. The three robots each have their own goals. Fig.1 shows the position and angles of robot2 and goal1 viewed from robot1.

The speed of each robot is $|V|$ and is set to be constant. The distance between robot1 and robot2 is denoted by D_{12}. The angle between the direction of robot1 and the direction of robot2 viewed from robot1 is denoted by δ_{12}. In the same way, the angle between the direction of robot1 and the goal is θ_1. V_{R12} is the relative velocity of robot2 viewed from robot1. The angle between the relative velocity V_{R12} and the direction from robot1 to robot2 is ϕ_{12}. When ϕ_{12} is nearly zero, robot2 is approaching robot1 head on. Each angle is set to be counterclockwise positive when viewed from the reference line in Fig.1.

3. Fuzzy classifier system

The CS, Goldberg (1989), is a learning system that has the following four components: a rule generation mechanism, a rule base, a production system and

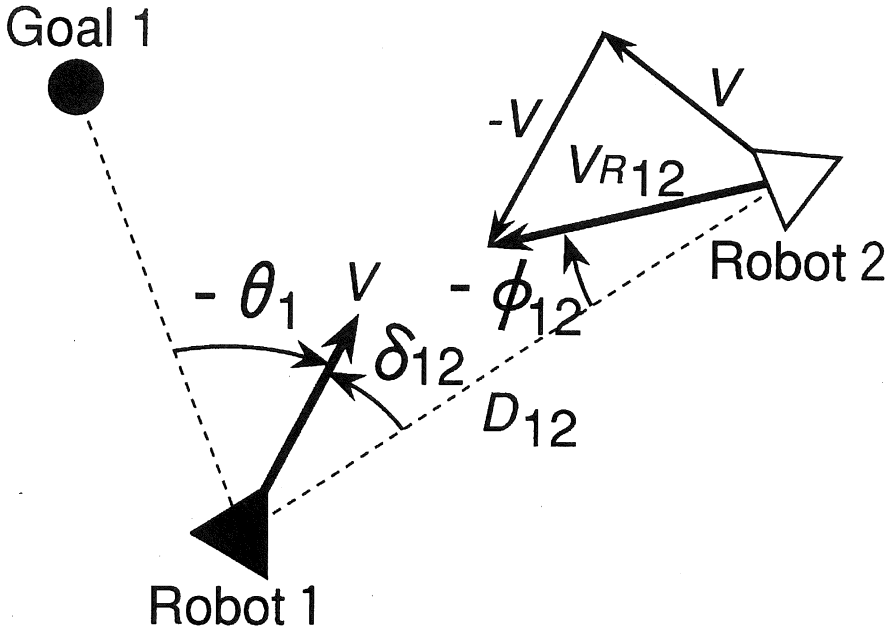

Figure 1. Parameters of the robot

an apportionment of credit system. The reinforcement learning using the CS is done in an environment where the movable space of the robot is crisply divided into small subspaces. The information of the obstacles and the goal, as well as the actions of the robot are handled as discrete values. Action selection among several active production rules should also be done carefully.

The FCS replaces the rule base by a fuzzy rule base, and the production system by a fuzzy inference system. The configuration of the FCS is shown in Fig.2. The fuzzy inference system can handle continuous variables and can aggregate all the activated rules easily. On the other hand, the apportionment of credit system should be carefully designed to assign credits to multiple rules which were active at the same instance. In this paper, robot1~robot3 are equipped with FCS1~FCS3, respectively. Each robot is expected to find control knowledge through the success and failure of its actions. However, it is difficult to specify the cause of the success or failure in the case where the three robots change fuzzy rules simultaneously. When one robot is learning, the fuzzy rules of the two other robots are fixed. The two other robots become the moving obstacles for the learning robot. Thus the learning of the three robots is done alternately.

Considering the case where the robot1 finds the fuzzy rules, the functions of the components of FCS1 are explained as follows:

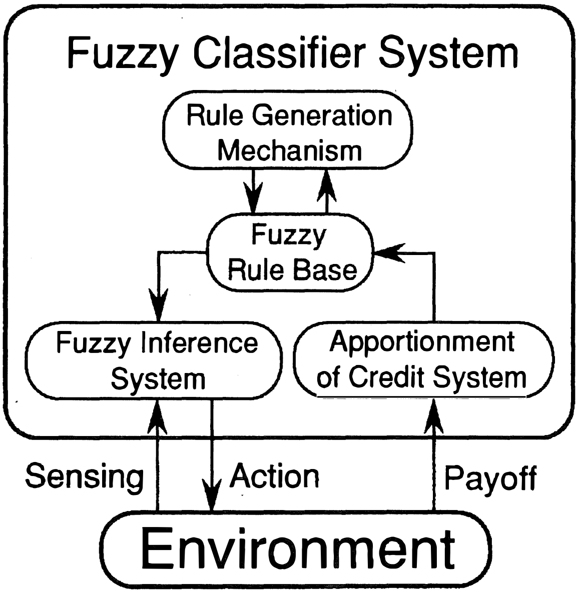

Figure 2. Configuration of the fuzzy classifier system

3.1. Fuzzy rule base

The inputs of the FCS1 are the relative distances D_{12}, D_{13}, the angles ϕ_{12}, ϕ_{13}, the relative angles δ_{12}, δ_{13}, and the relative angle θ_1. The output of FCS1 is the steering angle u_1. Two membership functions (Small (S), Big (B)) are used for the distances D_{12}, D_{13}, and five (Negative Big (NB), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Big (PB)) are used for other variables. A rule of FCS1 is encoded into eight loci as in Fig.3.

Each locus corresponds to one of the labels of the membership functions for each variable. The rule in Fig.3 can be read as follows:

D_{12}	D_{13}	ϕ_{12}	ϕ_{13}	δ_{12}	δ_{13}	θ_1	u_1
S	B	PS	NS	NS	PB	NB	PB

Figure 3. An example of coding of input/output variables

$$\text{IF } D_{12} \text{ is } S, D_{13} \text{ is } B, \phi_{12} \text{ is } PS, \phi_{13} \text{ is } NS, \delta_{12} \text{ is } NS, \delta_{13} \text{ is } PB, \\ \theta_1 \text{ is } NB \text{ THEN } u_1 \text{ is } PB. \tag{1}$$

Fig.4 shows the membership functions for each variable. The width of membership functions is determined through several trials.

3.2. Fuzzy inference system

Fuzzy inference is done here using fuzzy rules in the fuzzy rule base. The product-sum-center of gravity method is used. The steering of robot1 is continued until robot1 collides with either of the two other robots, or robot1 goes off the screen on the CRT, or robot1 reaches the goal avoiding collision with the two other robots.

3.3. Apportionment of credit system

In this paper, two kinds of methods are presented: (a) payoffs and credits based on the performance of the system, (b) payoffs and credits based on the number of the valid membership functions.

3.3.1. Payoffs and credits based on the performance of the system

Payoffs are given to the apportionment of credit system on success/failure of the steering. Success means that the robot reaches its goal avoiding collision with the two other robots, and failure means that the robot goes out of the screen, or the robot collides with one of the two other robots. Fig.5 shows the definition of payoffs. The FCS1 loaded on robot1 receives a positive payoff in case of success, and receives a negative payoff in case of failure. The feature of this system is that the FCS finds the rules using such a simple standard of evaluation, without being taught detailed rules by operators.

The way of delivering the credits α in accordance with the payoffs to each fuzzy rule in the rule base is as follows:

Each rule has the initial credit of α_0. Suppose that the simulation is at the k-th ($k = 1, 2, \cdots$) generation. The steering of robot1 is done n_{tri} times in a generation. The truth value and the output steering angle u_1 of each fuzzy rule at each sampling time from the start to the goal/failure are memorized. FCS1 receives the payoff $\mu_S(> 0)$ in case of success or the payoff $\mu_F(< 0)$ in case of failure. The apportionment of credit system assigns these payoffs to each fuzzy rule as the credit $\Delta\alpha_{jk}$ ($j = 1, 2, \ldots, n_{chr}$). n_{chr} means the number of rules. The mean values of the positive and negative steering angles u_1 for each steering trial are obtained separately. Then the rules which satisfy the following condition receive the credit given by (3).

$$|u_1| > \gamma_{thr}|\text{mean value of the steering angle}| \tag{2}$$

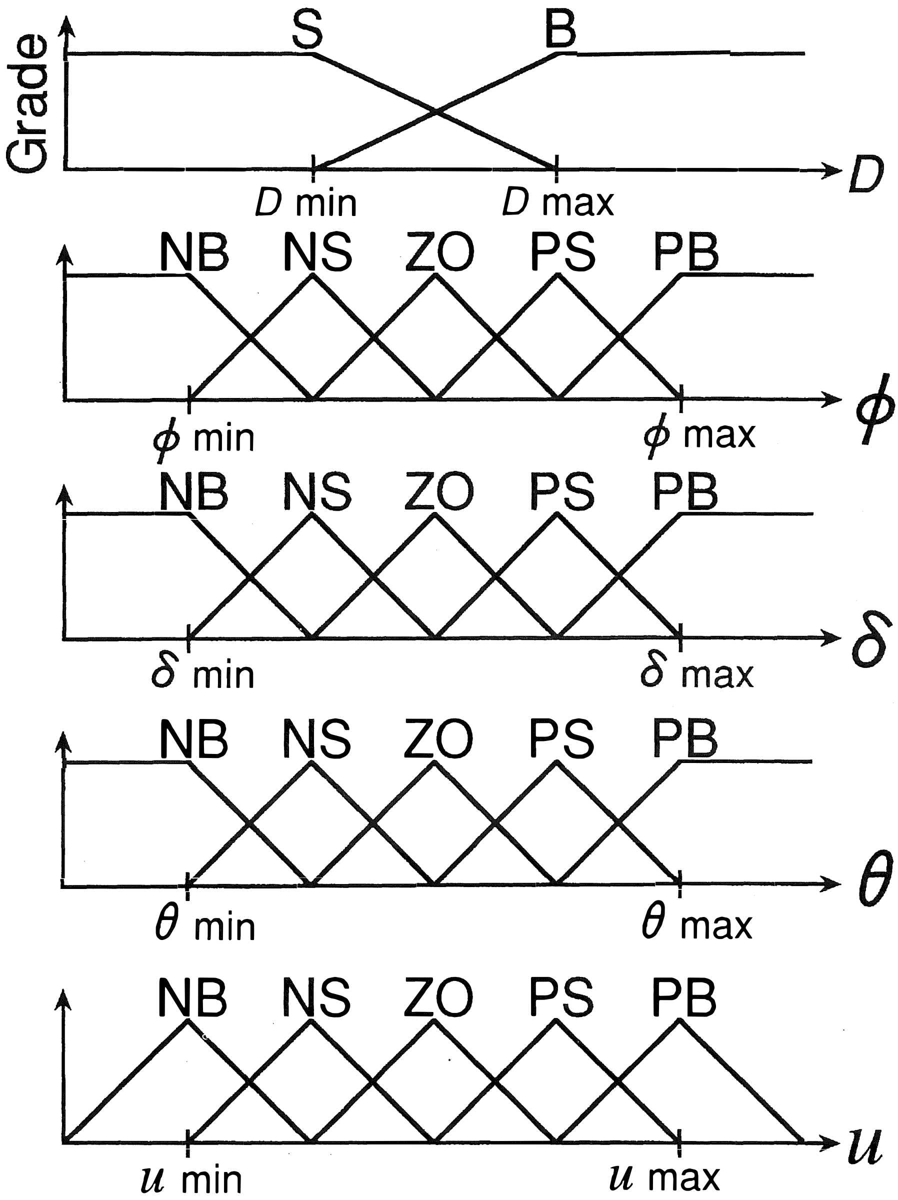

Figure 4. Membership functions

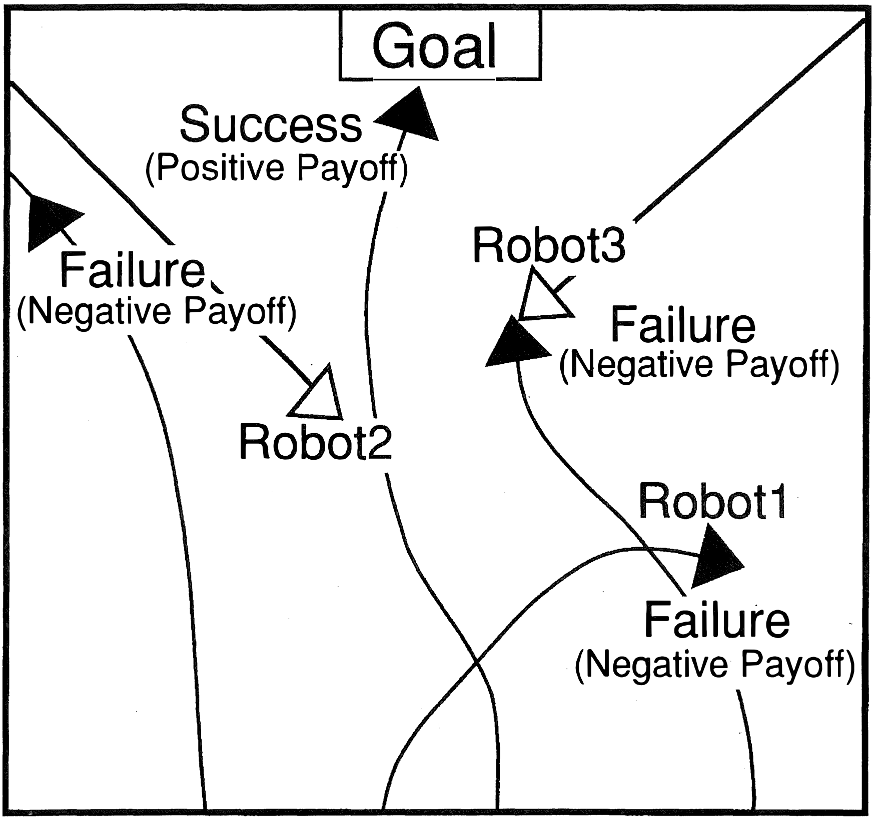

Figure 5. Definition of payoffs

$$\Delta\alpha_{jk} = \mu_i \sum (\text{the truth value of the fuzzy rule})_l$$
$$(i = S, F,\ j = 1, 2, \ldots, n_{chr}) \tag{3}$$

where $\gamma_{thr} \in (0, 1.5)$ determines the threshold level, l means the l-th sequence of events in a control simulation.

The rules with the output $|u_1|$ smaller than $\gamma_{thr} \times |$mean value of the steering angle$|$ receive no credit ($\Delta\alpha_{jk} = 0$). Thus the credits of the rules at the $(k+1)$-th generation α_{jk+1} is renewed as follows:

$$\alpha_{jk+1} = \alpha_{jk} + \Delta\alpha_{jk}. \tag{4}$$

The rules which have never had the truth value bigger than 0 pay the tax $\Delta\alpha_{tax}(< 0)$ as follows:

$$\alpha_{jk+1} = \alpha_{jk} + \Delta\alpha_{tax}. \tag{5}$$

The credit α, together with the crossover and mutation operators, is effective for finding valid rules for the steering.

3.3.2. Payoffs and credits based on the number of the valid membership functions

The number of possible rules $n_{pos} = 2^2 \times 5^6 = 62,500$ in this system, whereas most of these n_{pos} rules are anticipated not to be used for controlling the robot. It is easily expected that a smaller number of rules relative to n_{pos} is sufficient to fulfil the steering task. On the contrary, with the small number of randomly generated rules, the number of rules which have truth values larger than zero is expected to be very small, and the steering of the robot becomes difficult. It is hard to find rules which have truth values larger than zero with payoffs based on the performance of the system. Thus the payoffs and credits based on the number of valid membership functions are used for generating proper fuzzy rules. These payoffs are directly given to the fuzzy rules whose membership functions in the antecedent have a certain amount of grades at each sequence of events in the simulation. For every generation, e.g. the k-th ($k = 1, 2, \ldots$) generation, the credit β_{jk} ($j = 1, 2, \ldots, n_{chr}$) of each fuzzy rule is accumulated in proportion to the number of membership functions having grades larger than zero. At the early generations, there are few rules which have truth values. This credit β, together with the crossover operator, is effective for searching valid combinations of membership functions in the antecedent.

3.4. Rule generation mechanism

In this component, fuzzy rules are selected and reproduced using the genetic algorithm (GA). In this paper, the crossover and mutation operators are used.

3.4.1. Selection and reproduction

The n_{sel} rules with the least credits β_{jk+1} are selected. Among the selected n_{sel} rules, there may be some rules having large credits of α_{k+1}. Let ζ be the n_{less}-th value from the least credit α among the credits of chromosomes α_{k+1}. In the selected n_{sel} rules, the rules whose credits α_{k+1} are larger than ζ are replaced with the rules having less credits of α_{k+1}. The n_{sel} rules selected in this way are screened out and new n_{sel} rules are randomly reproduced from the remaining rules.

3.4.2. Crossover

The crossover operation is used especially for searching fuzzy rules which have truth values at each phase of the steering operations. One-point crossover is applied to the newly reproduced rules. Using the credit β, the rules which have valid combinations of membership functions can survive and the rules which can be activated are found by means of the crossover operation among the reproduced rules.

3.4.3. Mutation

The mutation operation, which changes the labels of membership functions, is used for the following two purposes: One is application to the antecedent parts of the newly generated rules in case the antecedent part of the new rules coincide with that of the new rules for avoiding rule uniformity. The other is application to the consequent parts of the existing rules with a probability of 0.5 for improving the actions.

After these genetic operations are done, the credits of the new rules and the credits of the rules whose credits are less than α_0 are renewed to be α_0, and the other credits of all rules β are reset to zero. This is to give the new rules an equal chance to survive. The rules at the $(k+1)$-th generation are produced and the simulation restarts from (2) fuzzy inference system.

The operations of FCS1 are explained above. Those of FCS2 and FCS3 are done in the same way. Each robot learns for n_{gene} generations alternately.

4. Simulations

The rules in each FCS were first randomly generated. The number of rules $n_{chr} = 100$, the number of trials $n_{tri} = 5$, initial credit $\alpha_0 = 100$, the payoff in case of success $\mu_S = 3.0$, the payoff in case of failure $\mu_F = -1.0$, the threshold of the steering angle $\gamma_{thr} = 1.5$, the tax $\Delta\alpha_{tax} = -0.1$, the number of rules which are screened out $n_{sel} = 20$, the number n_{less} which determines the threshold value ζ is set as $n_{less} = 30$, the generations for a robot to learn in sequence $n_{gene} = 30$. When all the robots reach the goal in every trial n_{tri} or the 900th generation is over, the simulation is stopped. Robot1$\sim$robot3 appear from the

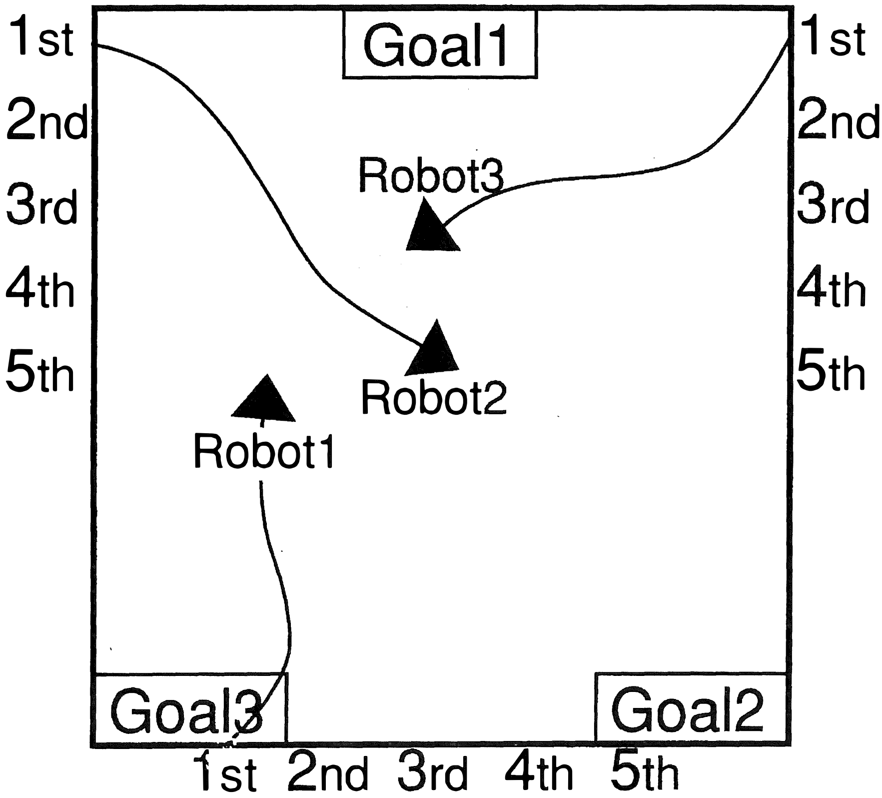

Figure 6. Locations of starting points and goals

bottom line, the left-hand side line and the right-hand side line, respectively. Each robot has 5 different starting points. The starting points and the goals of robot1~robot3 are located as shown in Fig.6. If the robots do not steer, the robots will collide with the other robots.

Fig.7 shows an example of the tracks of the robots with the initial fuzzy rules. All the robots went straight without steering because there existed few rules which had truth values. Fig.8 shows an example of the tracks of the robots at the 91st generation (after the three robots have learned for 30 generations each). The rules which have the truth values were produced in each FCS and each robot was steered a little. However, the steering was not sufficient to reach the goal avoiding collision with the two other robots yet. Figs.9-11 show examples of the tracks of the robots at the 650th generation. Figs.9-11 show the

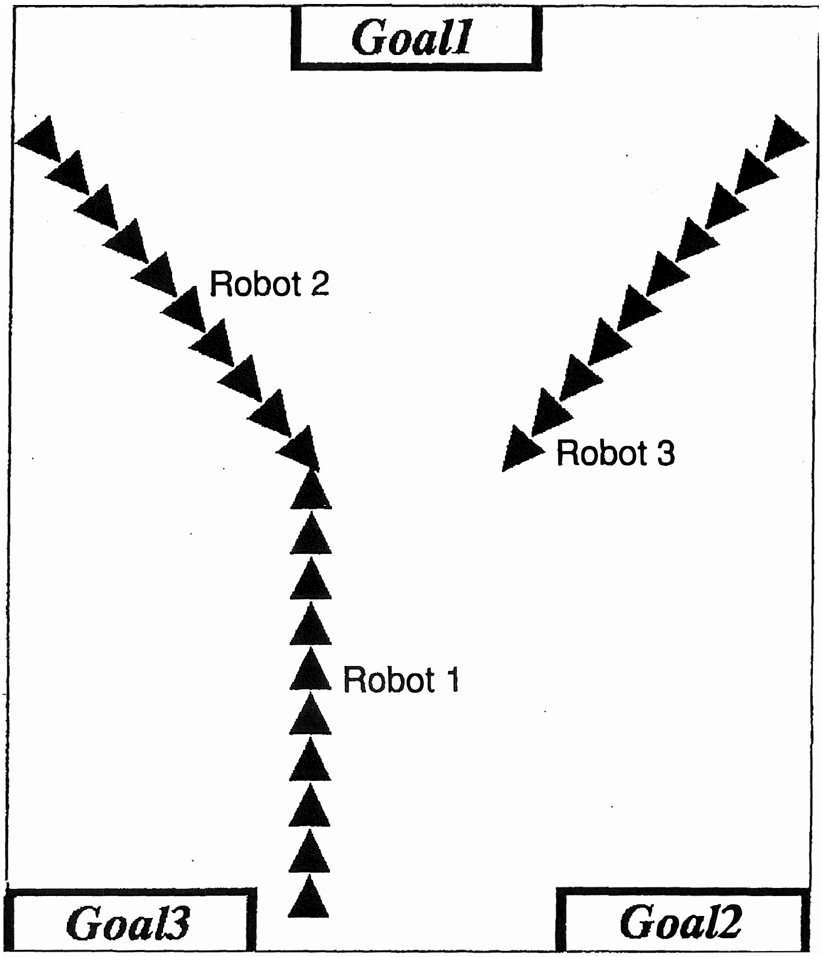

Figure 7. Tracks of robots with initial fuzzy rules

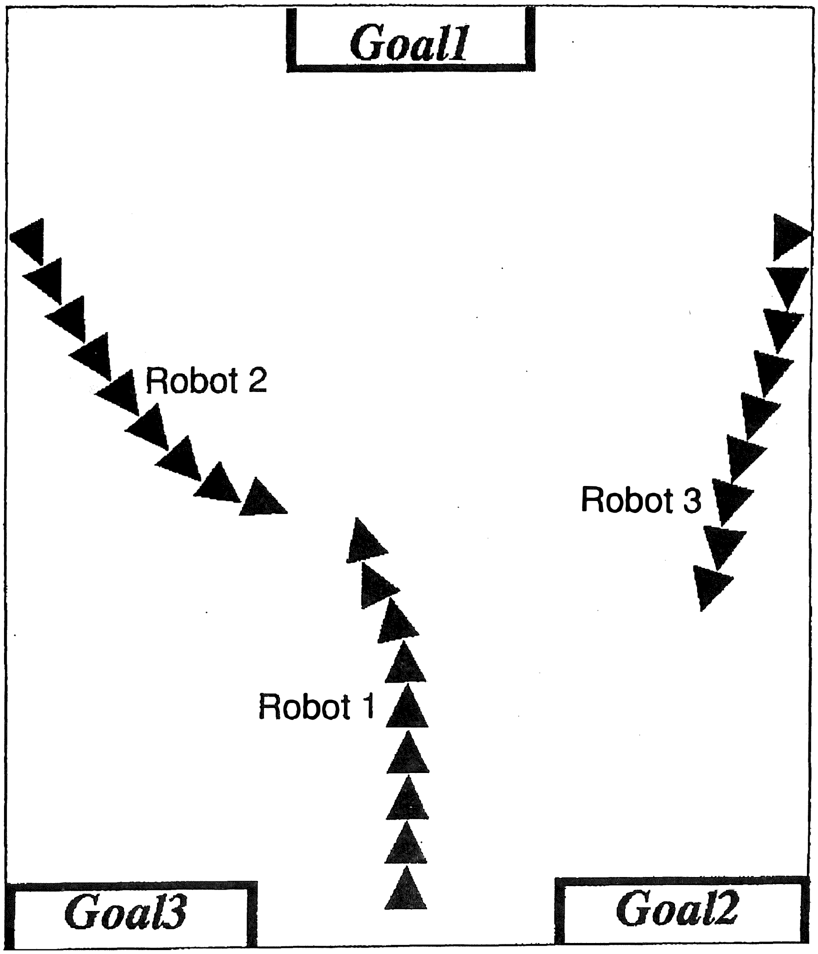

Figure 8. Tracks of robots at the 91st generation

cases where three robots started from the 1st, 4th, and 5th points respectively. With the improved fuzzy rules, all the robots steered well.

At the point 1 in Fig.10, the three robots used the fuzzy rules as follows:

Robot1

IF D_{12} is B, D_{13} is B, ϕ_{12} is PM, ϕ_{13} is PM, δ_{12} is PB, δ_{13} is PB,
θ_1 is PM THEN u_1 is NB.

Robot2

IF D_{23} is S, D_{21} is B, ϕ_{23} is PM, ϕ_{21} is PM, δ_{23} is ZO, δ_{21} is PB,
θ_2 is PM THEN u_2 is ZO.

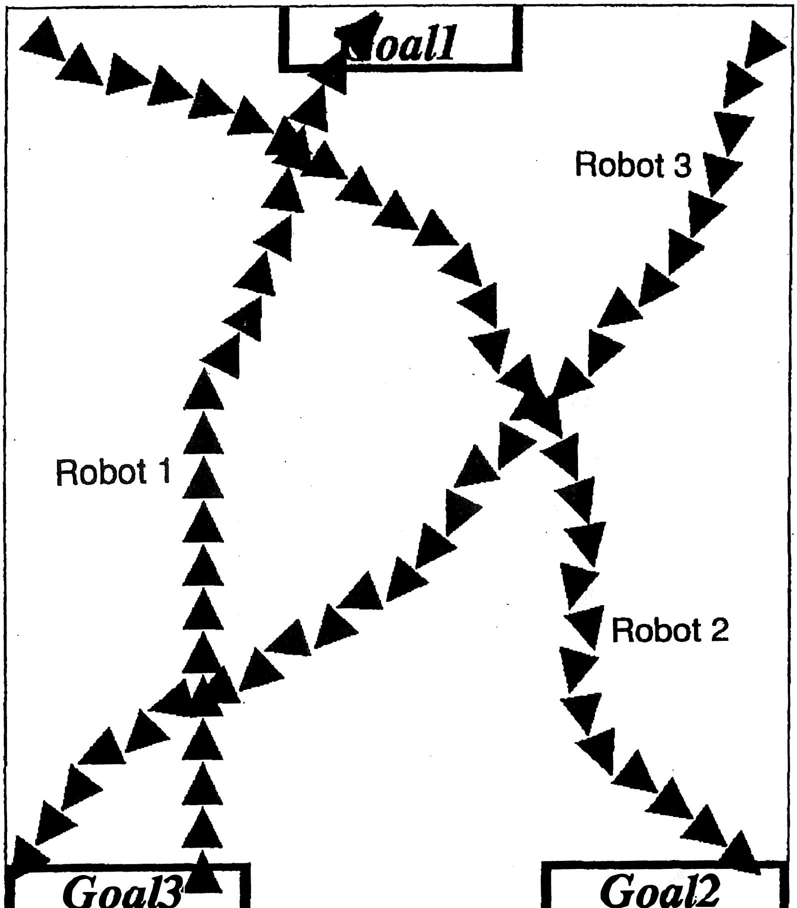

Figure 9. Tracks of robots which started from 1st points at the 650th generation

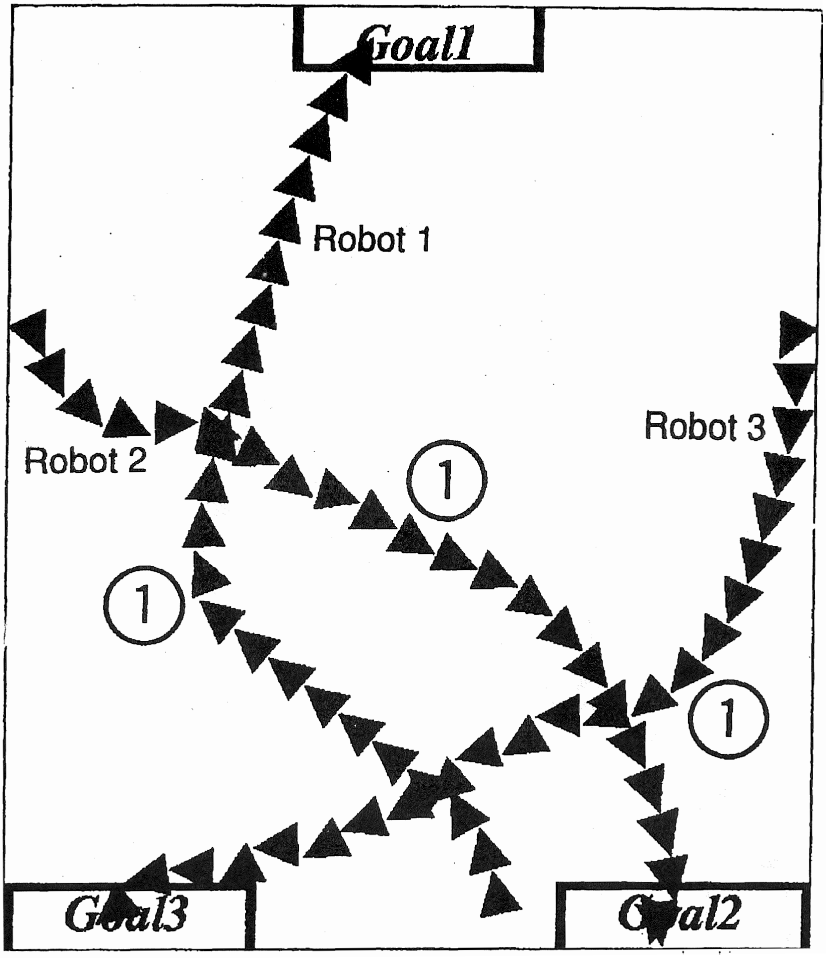

Figure 10. Tracks of robots which started from 4th points at the 650th generation

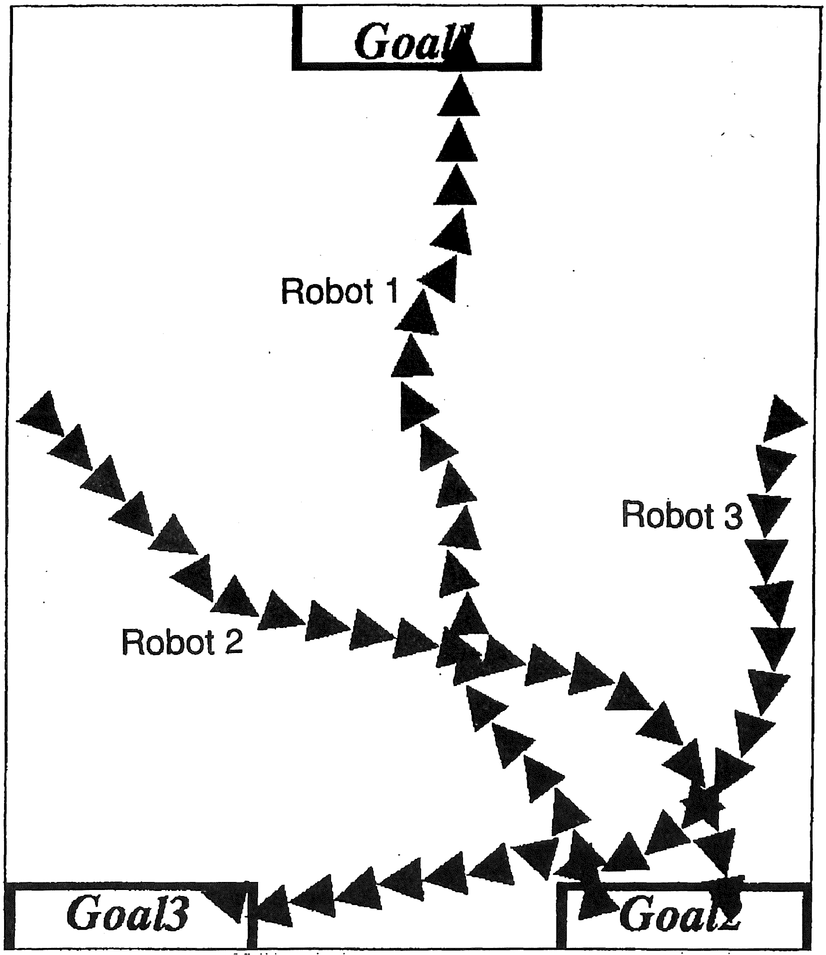

Figure 11. Tracks of robots which started from 5th points at the 650th generation

Robot3

IF D_{31} is B, D_{32} is S, ϕ_{31} is PM, ϕ_{32} is PM, δ_{31} is PM, δ_{32} is PM, θ_3 is ZO THEN u_3 is NB.

In order to investigate the generality of the obtained fuzzy rules, the robots were steered under the conditions different from the cases when the learning occured. Figs.12-14 show examples of the tracks of the robots with the fuzzy rules acquired at the 650th generation in the case where some robots started from other points. In Fig.12, the starting point of robot1 was the 4th, and those of robot2 and 3 were the 1st. In Fig.13, the starting point of robot1 was the 1st, and those of robot2 and 3 were the 5th. In Fig.14, the starting point of robot1 was the 3rd, that of robot2 was the 1st, and that of robot3 was the 5th. Under these conditions in which the three FCSs were not specifically trained, all the robots steered well to reach their goals.

Conclusions

This paper has detailed our attempts to find control knowledge using FCSs. New payoffs and credits were proposed. Simulation results demonstrated that the three mobile robots which were equipped with FCS controllers could find the fuzzy rules for avoiding collision among the robots. Because each robot moved according to its own fuzzy rules and fuzzy rules were found in each FCS separately, the simulated environment was complex and dynamic. Simulations meant to investigate the generality of the obtained fuzzy rules were also done.

References

FURUHASHI, T., NAKAOKA, K., MORIKAWA, K., UCHIKAWA, Y. (1993) Controlling excessive fuzziness in a fuzzy classifier system. *Proceedings of the Fifth International Conference on Genetic Algorithm*, 635.

FURUHASHI, T., NAKAOKA, K., MORIKAWA, K., UCHIKAWA, Y. (1994) An acquisition of control knowledge using multiple fuzzy classifier systems. *Journal of Japan Society for Fuzzy Theory and Systems*, **6**, 3, 603-609.

GOLDBERG, D.E. (1989) *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison Wesley.

IWASHITA, T., YAMAMURA, M., KOBAYASHI, S. (1994) Emerging cooperation by reinforcement learning. *Proceedings of the Eighth Annual Conference of JSAI*, 227-230.

NAGAYOSHI, M. (1994) An Experiment of Evolution of Hereditary Expedition Strategy by Reinforcement Learning. *Proceedings of the Nineteenth Intelligent System Symposium*, 7-10.

VALENZUELA-RENDON, M. (1991) The fuzzy classifier system: A classifier system for continuously varying variables. *Proceedings of the Fourth International Conference on Genetic Algorithm*, 346-353.

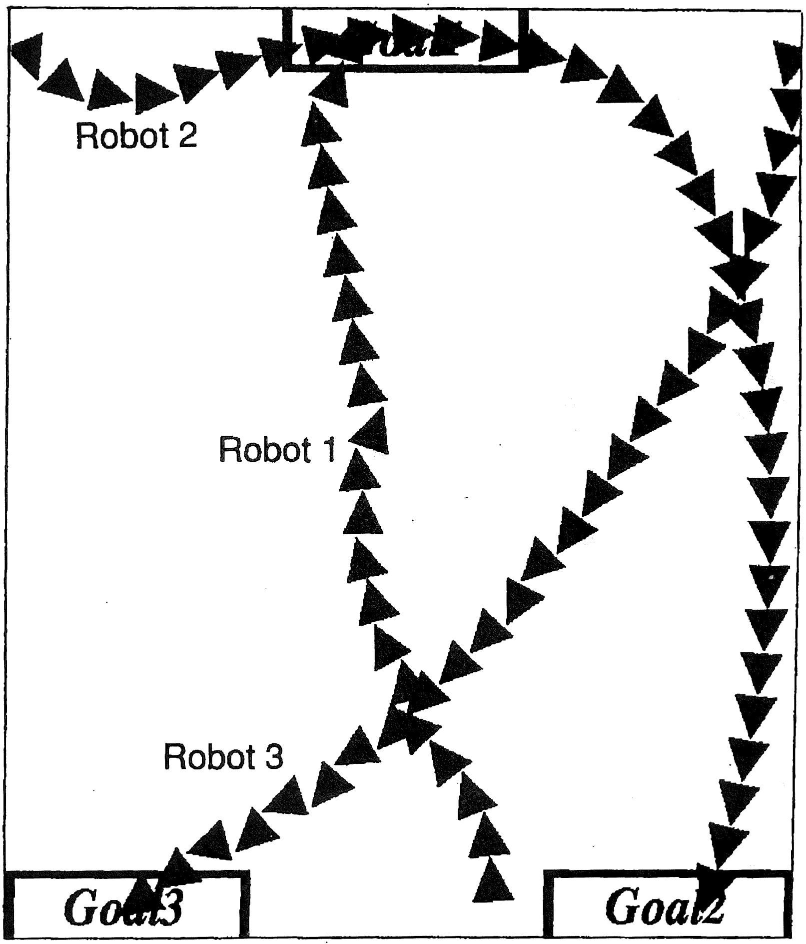

Figure 12. Tracks of robots which started from different points

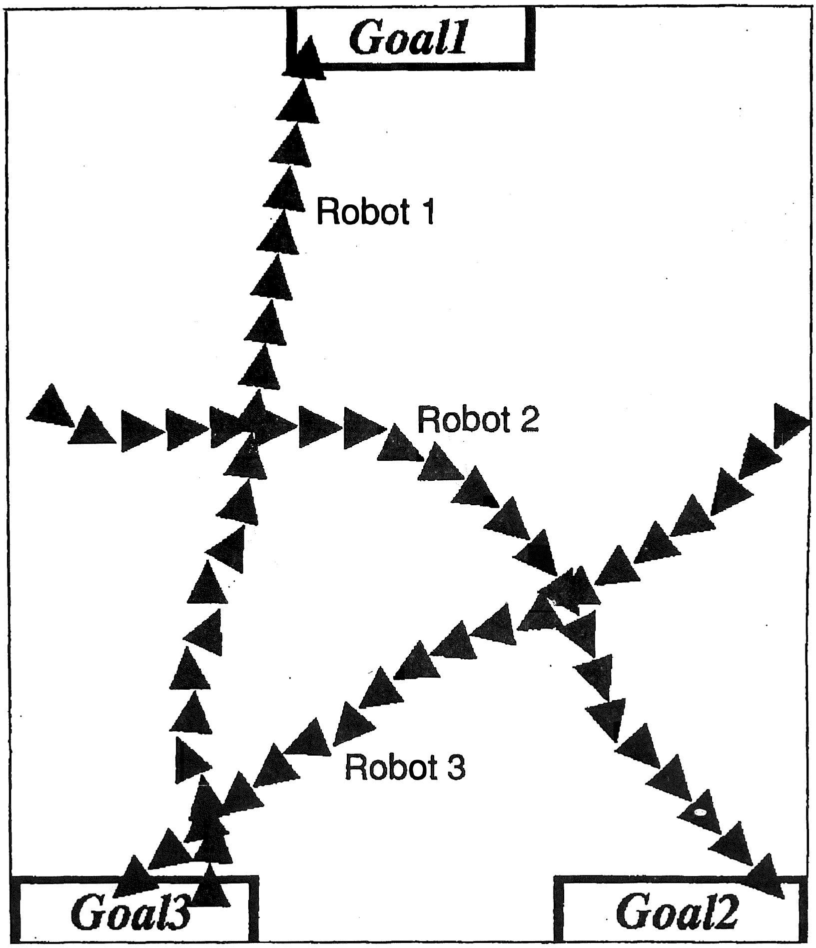

Figure 13. Tracks of robots which started from different points

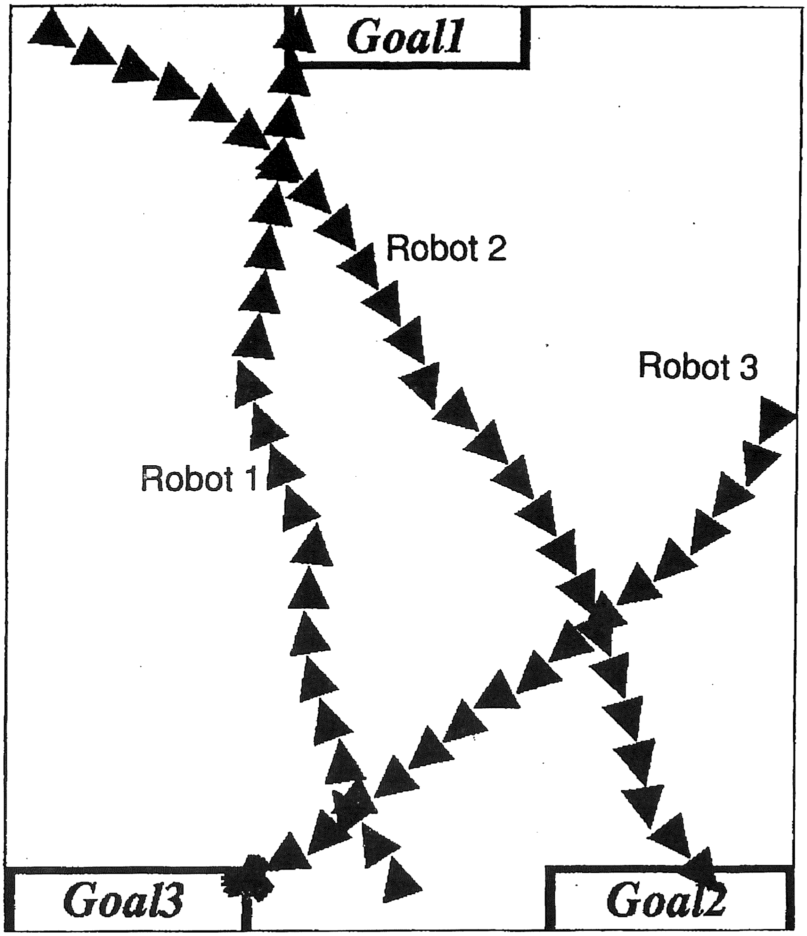

Figure 14. Tracks of robots which started from different points

Regular Papers