Proceedings of the 5th World Congress on Intelligent Control
and Automation, June 15-19, 2004, Hangzhou, P.R. China

第五届全球智能控制与自动化大会
2004 年 6 月 15 - 19 日，中国杭州

Intelligent Obstacle Avoidance for an Autonomous Mobile Robot

Xiaochuan Wang and Simon X. Yang
Advanced Robotics and Intelligent Systems (ARIS) Lab
School of Engineering, University of Guelph
Guelph, ON, N1G 2W1, Canada

Max Q.-H. Meng
Department of Electronic Engineering
Chinese University of Hong Kong
Shatin, N.T., Hong Kong

***Abstract*— In this paper, an intelligent obstacle-avoidance approach to autonomous navigation of a mobile robot in unknown environments is developed using neuro-fuzzy technique. A combination of four infrared sensors is equipped to detect the distance to obstacles around the mobile robot. The distance information is processed by the proposed neuro-fuzzy controller to adjust the velocities of the differential-drive system of the mobile robot. Simulation and experimental results demonstrate the effectiveness of the developed neuro-fuzzy controller.**

I. INTRODUCTION

Robot, especially mobile robot, is the common physical embodiment for problem solving, planning, and other areas of artificial intelligence. In recent years, autonomous mobile robots have been applied in various fields such as space and underwater exploration, industry and military usage. Obstacle avoidance is one of the most important issues for the navigation of autonomous mobile robots. Fuzzy logic and neural networks are two common ways to achieve obstacle avoidance. Fuzzy logic offers a possibility to mimic expert human knowledge. However, there lacks of a systematic methodology for conventional fuzzy system design. It is a time-consuming task to tune the parameters of each membership function. Neural networks have the capability to learn from existing knowledge. But the knowledge representation is very difficult. The integration of fuzzy logic and neural networks, neuro-fuzzy system, gives us a better way to realize obstacle avoidance. It inherits the advantages of both systems: fuzzy logic is used to treat ambiguous and imprecise information based on a set of linguistic rules; neural networks are used to tune the membership functions automatically so as to ensure better performance.

In the past two decades, many methods have been developed to realize obstacle avoidance action on mobile robots. Borenstein and Koren [9] proposed "histogramic in-motion mapping algorithm", which works well in a known environment. But it requires heavy computation and large memory. They also developed and implemented a method of virtual-force field for mobile robots [11]. However, they did not mention how to apply the theory to obstacle avoidance action. Akishita [7] presented an algorithm using Laplace potential theory. All these methods can only work well in a known environment.

Fuzzy logic and artificial neural works [10] are normally used for obstacle avoidance in unknown environments. Neuro-fuzzy model combines the advantages of both fuzzy logic system and neural networks. It keeps the simplicity of fuzzy system while endowing the system with learning capability. Godjevac [4] proposed a neuro-fuzzy model for a mobile robot to avoid obstacles. But more than 600 basic rules are used while many of them are redundant. The method of how to suppress useless rules is not introduced. In addition, the physical meanings of membership functions have been lost during the training.

In this paper, a novel neuro-fuzzy control model is presented to realize obstacle avoidance navigation for a indoor mobile robot developed in our ARIS Lab. Using the proposed neuro-fuzzy control model, the mobile robot can partially "see" the surrounding unknown environment and explore the unknown environment without hitting any static or slowly-moving obstacles automatically.

II. THE MOBILE ROBOT AND ITS KINEMATIC MODEL

A mobile robot (Fig. 1) is developed for testing the proposed neuro-fuzzy controller. The mobile robot is composed by three parts: the locomotion system, the sensor system and the programmable reasoning (control) system, as shown in Fig. 2. All these system are based on a $20 \times 25c$ platform.

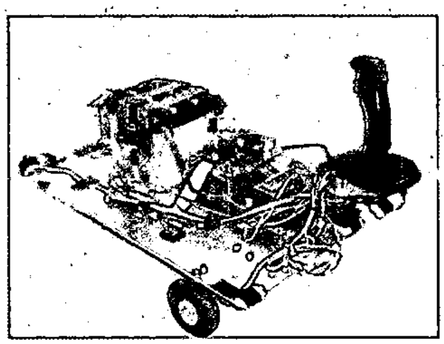

Fig. 1. The mobile robot.

Differential drive is used for the locomotion of the robot. It has two co-axle fixed wheels driven by different motors separately, and a third passive omni-directional caster to keep balance. Through adjusting the velocity of the two driven wheels respectively, the velocity and motion direction of the mobile robot can be determined. The kinematic model of

0-7803-8273-0/04/$20.00 ©2004 IEEE

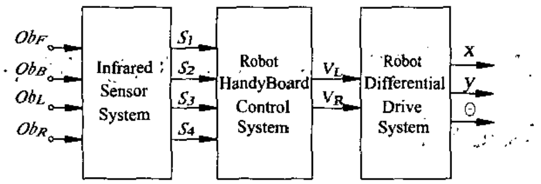

Fig. 2. The control architecture of the mobile robot.

the drive system is shown in Fig. 3. At any time, it can be described as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} 0.5c & 0.5c \\ 0.5s & 0.5s \\ -1/L & 1/L \end{pmatrix} \begin{pmatrix} V_L \\ V_R \end{pmatrix}, \quad (1)$$

where $\dot{x}$, $\dot{y}$ and $\dot{\theta}$ are the velocities in X, Y direction and the angular velocity of the cart, and V_L/V_R are the velocities of the two driven wheels.

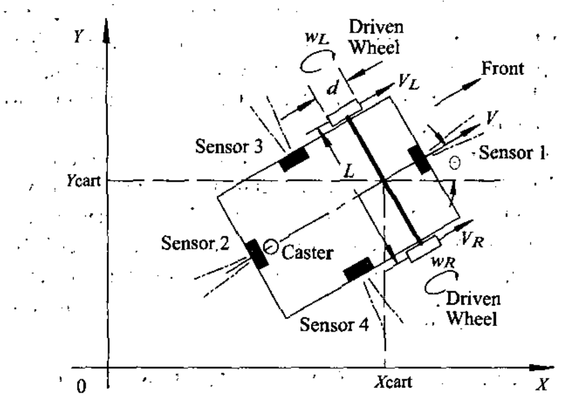

Fig. 3. The kinematic model of the mobile robot.

To detect the obstacles around effectively, a combination of 4 SHARP GP2D02 infrared sensors is used to facilitate navigation and obstacle avoidance. The sensors are located on the four sides of the cart to detect the obstacles around (Fig. 3). The effective detection scope for these sensors is approximately 10cm to 80cm. The relationship between the distance and the digital sensor value (S_i) is shown in Fig. 5. The sensor data are obtained through an 8-digit AC/DC sampler before they are processed by the microprocessor. So the data range is from 0 to 255. Here the supplementary part of sensor data are used as the sensor input I_i of the neuro-fuzzy controller, i.e.,

$$I_i = 255 - S_i, \quad i = 1, 2, 3, 4, \quad (2)$$

where i is the index of the sensors. Thus I_i increases when the distance increases in the detection range. This figure is used in the programming and computer stimulation to train the parameters of the obstacle avoidance controller.

Handy Board is applied on this mobile robot as the central control system. The Handy Board is a hand-held, battery-powered microcontroller board developed by MIT. Based on the Motorola 68HC11 microprocessor, 32K RAM, it can be programmed by language of Interactive C and execute some certain tasks including receiving data from sensors, reasoning and driving the DC motors. It is ideal for this mobile robot.

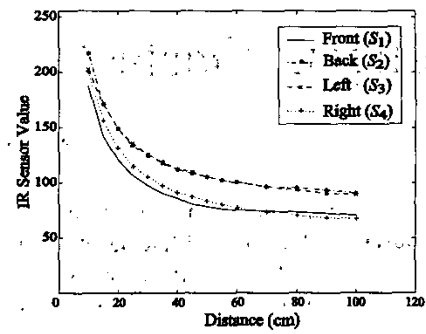

Fig. 4. The relationship between the digital sensor value and the distance to the obstacle.

III. THE NEURO-FUZZY CONTROLLER FOR OBSTACLE AVOIDANCE

In this section, the neuro-fuzzy model for obstacle avoidance of the mobile robot is firstly presented. Then the fuzzification, the linguistic rule base, the fuzzy inference mechanism and the defuzzification of the proposed neuro-fuzzy model are introduced. At last, the learning algorithms for the neuro-fuzzy model are given.

A. The Neuro-fuzzy Model

To realize obstacle avoidance, four MISO neuro-fuzzy controllers in parallel are used (Fig. 5). The inputs are the readings of the infrared sensors (I_{1-4}), the outputs are the sub-velocities (V_{L1}, V_{L2}, V_{R1} and V_{R2}) of the driven wheels. Then the velocities of left and right wheels, $V_L = V_{L1} + V_{L2}$ and $V_R = V_{R1} + V_{R2}$, are applied to the motion control of the mobile robot.

The four neuro-fuzzy controllers have the same architecture except different linguistic rule bases. So in the following part we can use Neuro-fuzzy Controller 1 as an example.

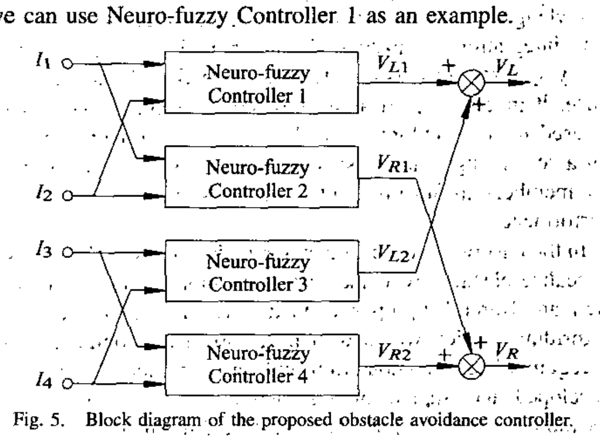

Fig. 5. Block diagram of the proposed obstacle avoidance controller.

B. Fuzzification

The schematic architecture of Controller 1 is shown in Fig. 6. This controller consists of a four-layer MISO neural networks constructed with the fuzzy logic algorithm. The first layer is the input layer. There are two neurons (I_1 and I_2) in this layer. Then the second layer (Hidden Layer 1) fuzzifies each input into three membership functions. There are 8 neurons in this layer. The third layer with 16 neurons (Hidden Layer 2) performs the inference mechanism on the base of fuzzy rules. The result can be obtained from the output layer through defuzzification.

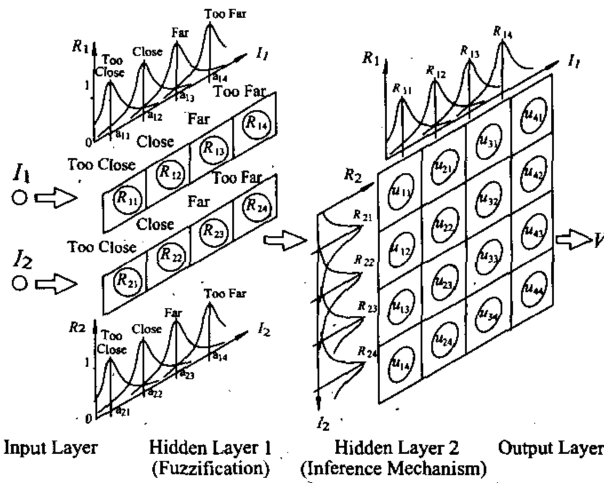

Fig. 6. Schematic architecture of Controller 1.

The fuzzification process is to map the crisp input values into the linguistic fuzzy terms with the membership functions between 0 and 1. In this paper, Gaussian functions are used to represent the fuzzy membership function.

C. Linguistic Rule Base

Four membership functions are defined for each infrared sensor according to the distance to the obstacles (shown in Fig. 7). The definition of membership functions is given as

$$R_{ij} = \exp\left(-\frac{(I_i - a_{ij})^2}{2\sigma_{ij}^2}\right), \quad i = 1,2,3,4; j = 1,2,3,4. \tag{3}$$

The obstacle detection is defined on four levels - Too Close (TC), Close (C), Far (F) and Too Far (TF). The linguistic rule base is the combination of a set of "IF-THEN" rules. To achieve the obstacle avoidance, 32 rules are defined in Tables I and II for the front/rear infrared sensors I_1/I_2 and left/right infrared sensors I_3/I_4 respectively. For example, the first rule in Table I can be described as: IF I_1 is Too Close and I_2 is Too Close, THEN V_{L1} is Me and V_{R1} is Me .

From the rule bases mentioned above, we can find that I_1/I_2 and I_3/I_4 have different contributions to the final velocities of left and right wheels, V_L and V_R. In Table I, the outputs of V_{L1} and V_{R1} are mainly related to the distance to the front obstacles. If there is obstacle lies on the front of the robot, the robot will turn a certain angle to avoid the obstacle. The shorter the distance, the larger the turning angle. In Table II, if there is an obstacle on the left/right of the robot, the robot will turn to the opposite direction to avoid the obstacle. The

TABLE I

FUZZY RULE BASE 1 FOR I_1 AND I_2. Z: ZERO; S: SLOW SPEED; M: MEDIUM SPEED; F: FAST SPEED; FF: FASTEST SPEED.

V_{L1}/V_{R1}			I_2			
			R_{21}	R_{22}	R_{23}	R_{24}
			TC	C	F	TF
I_1	R_{11}	TC	M/M	M/M	M/M	M/M
	R_{12}	C	S/M	S/M	S/M	S/M
	R_{13}	F	S/F	S/F	S/F	S/F
	R_{14}	TF	Z/FF	Z/FF	Z/FF	Z/FF

TABLE II

FUZZY RULE BASE 2 FOR I_3 AND I_4. Z: ZERO; S: SLOW SPEED; M: MEDIUM SPEED; F: FAST SPEED.

V_{L1}/V_{R1}			I_4			
			R_{41}	R_{42}	R_{43}	R_{44}
			TC	C	F	TF
I_3	R_{31}	TC	Z/Z	S/Z	M/Z	F/Z
	R_{32}	C	Z/S	Z/Z	S/Z	M/Z
	R_{33}	F	Z/M	Z/S	Z/Z	S/Z
	R_{34}	TF	Z/F	Z/M	Z/S	Z/Z

output velocity of V_{L2} and V_{R2} depends on the difference between I_3 and I_4. When the difference becomes larger, the difference between the output velocities of V_{L2} and V_{R2} becomes larger too. If some rules in Tables I and II are fired simultaneously, the combination of the motion will let the robot go forward/backward while turning at a certain angle so as to avoid the obstacles on each direction.

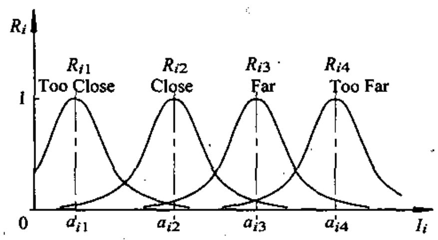

Fig. 7. Input membership functions.

D. Fuzzy Inference Mechanism

There are many algorithms for fuzzy inference. Normally the Mamdani arithmetic rule (minimum operation) is used. But this algorithm is not applicable to our study, because the derivatives of the functions do not exist, which are required for the training of the proposed neuro-fuzzy system. Since some important parameters in the derivatives are lost in the minimum operation, the neural network cannot be learned quickly. Here the Larson arithmetic rule – the product operator is used. So the firing strength of each rule can be expressed as

$$u_{ij} = R_{1i}R_{2j}, \quad i = 1,2,3,4; j = 1,2,3,4. \tag{4}$$

E. Defuzzification

The defuzzification process maps the output from the fuzzy inference mechanism to a crisp value. Normally there are two common methods for defuzzification - 'centroid of area' and 'mean of maximum'. The method of 'centroid of area' is more suitable in the proposed neuro-fuzzy controller, which combines the outputs represented by the implied fuzzy sets from all rules and generates the gravity centroid of the possibility distribution for a control action. Thus the crisp value of output can be obtained. In this case, the evaluation of the velocity of V, including V_{L1}, V_{L2}, V_{R1} and V_{R2}, is given by

$$V = \frac{\sum_{i=1}^{3}\sum_{j=1}^{3} u_{ij} w_{ij}}{\sum_{i=1}^{3}\sum_{j=1}^{3} u_{ij}}. \tag{5}$$

F. Learning Algorithms

The least mean square method and back propagation method are used to train the neuro-fuzzy system. In the neuro-fuzzy controller, the following vector of parameters needs to be tuned during training

$$\begin{aligned} z = \{ & a_1, \cdot\ a\ 1\ a\ 2\ 4\ \ a\ 2\ \sigma\ 1\ 4\ \ \sigma\ 1,\ \ 4 \\ & \sigma_2, \cdot\ \ \sigma\ 2\ w\ 1\ 4\ \ w\ 4\ \}. \ \ 4 \end{aligned} \tag{6}$$

The cost function is defined as

$$E(z) = \frac{1}{2}(V - V_T)^2 = \frac{1}{2}\Delta V^2, \tag{7}$$

where V is the output velocity of neuro-fuzzy system and V_T is the target velocity. The iterative procedure is defined as

$$z^{(n+1} = z^{(n)} - \eta\frac{\partial}{\partial\ ^{(n)}}, \tag{8}$$

where n is the iteration times, and η is the learning rate. So the equations for adaptation of the model parameters are given as

$$\begin{aligned} a_{1i}^{(n+1} &= a_{1i}^{(n)} - \eta_a \frac{\partial}{\partial\ _{1i}^{(n)}} \\ &= a_{1i}^{(n)} - \eta_a \frac{\Delta V^{(n)}(I_1 - a_{1i}^{(n)})R_{1i}^{(n)}A_2^{(n)}}{(\sum_{i=1}^{4}\sum_{j=1}^{4} u_{ij})(\sigma_{1i}^{(n)})^2}, \end{aligned} \tag{9}$$

$$\begin{aligned} a_{2j}^{(n+1} &= a_{2j}^{(n)} - \eta_a \frac{\partial}{\partial\ _{2j}^{(n)}} \\ &= a_{2j}^{(n)} - \eta_a \frac{\Delta V^{(n)}(I_2 - a_{2j}^{(n)})R_{2j}A_1^{(n)}}{(\sum_{i=1}^{4}\sum_{j=1}^{4} u_{ij})(\sigma_{2j}^{(n)})^2}, \end{aligned} \tag{10}$$

$$\begin{aligned} \sigma_{1i}^{(n+1} &= \sigma_{1i}^{(n)} - \eta_\sigma \frac{\partial}{\partial\ _{1i}^{(n)}} \\ &= \sigma_{1i}^{(n)} - \eta_\sigma \frac{\Delta V^{(n)}(I_1 - a_{1i}^{(n)})^2 R_{1i}^{(n)}A_2^{(n)}}{(\sum_{i=1}^{4}\sum_{j=1}^{4} u_{ij})(\sigma_{1i}^{(n)})^4}, \end{aligned} \tag{11}$$

$$\begin{aligned} \sigma_{2j}^{(n+1} &= \sigma_{2j}^{(n)} - \eta_\sigma \frac{\partial}{\partial\ _{2j}^{(n)}} \\ &= \sigma_{2j}^{(n)} - \eta_\sigma \frac{\Delta V^{(n)}(I_2 - a_{2j}^{(n)})^2 R_{2j}A_1^{(n)}}{(\sum_{i=1}^{4}\sum_{j=1}^{4} u_{ij})(\sigma_{2j}^{(n)})^4}, \end{aligned} \tag{12}$$

$$\begin{aligned} w_{ij}^{(n+1} &= w_{ij}^{(n)} - \eta_w \frac{\partial}{\partial\ _{ij}^{(n)}} \\ &= w_{ij}^{(n)} - \eta_w \frac{u_{ij}^{(n)}\Delta V^{(n)}}{\sum_{i=1}^{4}\sum_{j=1}^{4} u_{ij}^{(n)}}, \end{aligned} \tag{13}$$

where

$$\Delta V^{(n)} = V^{(n)} - V_T, \tag{14}$$

$$A_2 = \sum_{j=1}^{4}[(w_{ij} - V)R_{2j}], \tag{15}$$

$$A_1 = \sum_{i=1}^{4}[(w_{ij} - V)R_{1i}], \tag{16}$$

and η_a, η_σ and η_w are the learning rates for parameters training.

The training process stops on either of two conditions: the cost function E is smaller than the specified tolerance ϵ, or the program finished the preset iterations (t). The training procedure can be generalized as follows:

BEGIN
  Initialization Procedure;
    Set w_{ij} from 0 to 1 randomly;
    Set a_{ij} according to Fig. 5;
    Set σ_{ij} with the same value.
  DO (Training Procedure)
    Data input (I_{1-4});
    Fuzzy inference;
    Data output;
    Back propagation.
  UNTIL (E or t);
END

IV. EXPERIMENTS

In order to demonstrate the effectiveness of the proposed neuro-fuzzy controller in the real world, experiments have been conducted through three steps: first, programming the source code of the neuro-fuzzy controller and training the parameters on computer; second, simulating the controller in the virtual environment on PC to check the performance; At last, applying the modified program on the Handy Board on the real robot to control its navigation. Here off-line supervised learning is executed. The training input data and testing input data are obtained from the four infrared sensors in real environment. Desired outputs are derived from the expert's opinion. There are 100 groups of training data and 56 groups of testing data for each controller. According to the experiments, when the iteration number is around 30,000, we can get the best training result on computer.

When the environment changes, the characteristic curves of infrared sensors will change. We can modify the training and testing data and repeat the training process. Thus the mobile robot can adapt to new environment.

To prove the effectiveness of the control model, several tests have been simulated on computer. Here some approximations have been applied to simplify the simulation. First, the surface roughness of all obstacles and walls are 100%, thus, the value of infrared sensor is only related to the distance to obstacles no matter what reflection angle is. Second, the detection angle of infrared sensors is ignored. Third, All obstacles are considered as polygons.

In Fig. 8, the mobile robot is placed in a room ($10m \times 10m$) with some static obstacles. The mobile robot shows enough flexibility to avoid all obstacles. In Fig. ??, one of the obstacles can move slowly back and forth. The mobile robot can still detect it (at Point A and B) and avoid it successfully.

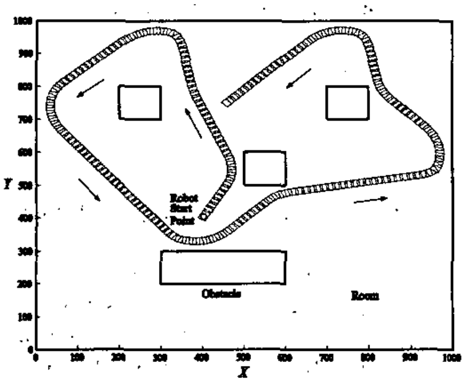

Fig. 8. Test 1: Robot navigation in a static environment.

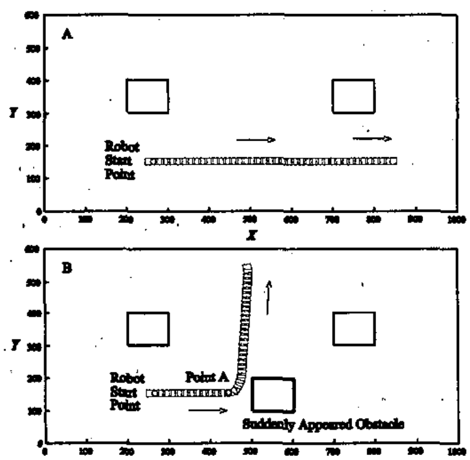

Fig. 9. Test 2: Robot navigation with sudden change. A: No obstacle on the path; B: Obstacle appears on the planning path suddenly when the robot reaches Point A

At last the neuro-fuzzy controller is applied on the real mobile robot. The mobile robot demonstrates satisfactory ability to avoid obstacles in the navigation autonomously. According to the distance between the robot and the obstacles, the robot took different obstacle avoidance actions (go in straight, turning a little bit to avoid the obstacle, or make a sharp turn. It can navigate autonomously in a room while avoiding bumping into any obstacles.

Besides, the comparison between this neuro-fuzzy controller, the conventional fuzzy logic obstacle avoidance controller and conventional neuro-fuzzy controller mentioned in [4] has also been done on this mobile robot. All of these controllers can get satisfying performance. But before the conventional fuzzy controller works well, all the parameters have to be trailed by time-consuming manual testing, there is no systematic way to get these values. Compared with the conventional neuro-fuzzy obstacle avoidance controller proposed in [4], the proposed controller use less sensors (4 v.s. 7) and simpler rules (32 v.s. 625) to realize obstacle avoidance function. The neuro-fuzzy controller can prevent over-fitting during training, because the redundancy of the fuzzy rule base has been reduced remarkably. Moreover, it has quicker time response because the fuzzy rules base and membership functions are much simpler. In other words, the proposed neuro-fuzzy controller performs better in a real-time control system.

V. Conclusion

In this paper, a new neuro-fuzzy controller for a mobile robot is developed and tested successfully. Compared to the fuzzy logic controllers, it provides a systematic method to obtain suitable membership functions through an automatic learning process, so that the time-consuming trial and error can be avoided. It can adapt to different environment through automatic learning process. Compared to other neuro-fuzzy controllers presented in [4], the proposed controller use less sensors and simpler rules to realize obstacle avoidance function, so that it performs better in a real-time control system. The proposed controller is applied to a mobile robot in the ARIS Lab and the effectiveness of the proposed method is demonstrated through experimental studies.

References

[1] X. Ma, X. Li,and H. Qiao, "Fuzzy Neural Network-based Real-time Self-reaction of Mobile Robot in unknown Environments," in *Mechatronics*, vol. 11, 2001, pp. 1039-1052.

[2] A. Fukayamaand, M. Ida, and O. Katai, "Behavior-based Fuzzy Control System for a Mobile Robot with Environment Recognition by Sensory-motor Coordination," in *Proceeding of 8th International Fuzzy Systems Conference*, Seoul, South Korea, 1999, pp. 105-110.

[3] Y. Shi and M. Mizumoto, "A New Approach of Neuro-Fuzzy Learning Algorithm for Tuning Fuzzy Rules," in *Fuzzy Sets and Systems*, 112, 2000, pp. 99-116.

[4] J. Godjevac, "A Learning Procedure for a Fuzzy System: Application to Obstacle Avoidance," in *Proceedings of the International Symposium on Fuzzy Logic*, Zurich, 1995, pp. 142-148.

[5] K. C. Ng and M. M. Trivedi, "A Neuro-fuzzy Controller for Mobile Robot Navigation and Multirobot Convoying," in *IEEE transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 28, no. 6, 1998, pp. 829-840.

[6] W. Li, C. Ma and F. M. Wahl, "A Neuro-fuzzy System Architecture for Behavior-based Control of a Mobile Robot in Unknown Environments," in *Fuzzy Sets and Systems*, vol. 87, 1997, pp. 133-140

[7] S. Akishita, T. Hisanobu and S. Kawamura, "Fast path planning available for moving obstacle avoidance by use of Laplace potential," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, Jul. 1993, pp. 673-678

[8] S. S. Haykin, *Neural Networks : a Comprehensive Foundation*, Upper Saddle River, N.J.: Prentice Hall, 1999

[9] J. Borenstein and Y. Koren, "Histogramic in-motion mapping for mobile robot obstacle avoidance," in *IEEE Transactions on Robotics and Automation*, vol. 7, no. 4, Aug, 1991, pp. 535-539.

[10] M. Zhang and S. Peng, "New approach for mobile robot obstacle avoidance based on multi-sonar information," in *Automatica Sinica*, vol. 24, no. 5, 1998, pp. 671-674.

[11] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," in *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 5, Sept-Oct, 1989, pp. 1179-1187.