Proceedings of the International Conference on Neural Information Processing, Hong Kong, 1996, pp. 948-953

An Intelligent Navigator for Mobile Vehicles

N. H. C. Yung and C. Ye

Department of Electrical and Electronic Engineering
The University of Hong Kong
Pokfulam Road, Hong Kong
Tel: (852)28592685 Fax: (852)28598738   Email: nyung@hkueee.hku.hk

***Abstract*—This paper presents an intelligent navigation method for navigation of a mobile vehicle in unknown environments. The proposed navigator consists of three modules: Obstacle Avoidor, Environment Evaluator and Navigation Supervisor. The Obstacle Avoidor is a fuzzy controller whose rule base is learnt through reinforcement learning. A new and powerful training method is proposed to construct the fuzzy rules automatically. The Navigation Supervisor determines the tactical requirement of avoiding obstacles or moving towards the goal location at each action step so that the vehicle can achieve its task without colliding with obstacles. The effectiveness of the learning method and the whole navigator are verified by simulation.**

1 Introduction

Path planning is an important issue in the navigation of mobile vehicles, which could be divided into two major categories: (1) global path planning, and (2) local path planning. Global path planning methods, such as roadmap [1], cell decomposition [2] are usually carried out in an off-line manner in a completely known environment. As a result, they are not suitable for navigation in complex, unknown or dynamic environments. On the other hand, local path planning techniques, also called obstacle avoidance methods which are carried out in an on-line manner are more efficient in the navigation of mobile vehicles in such environment.

Of the local path planning methods, potential field method [3, 4] seems quite efficient in obstacle avoidance, however it has two disadvantages: First, it is difficult to find the force coefficients influencing the velocity and direction of the mobile vehicle in a complicated environment, which are too complex to be embedded in a mathematical model. Second, the potential local minimum could cause the vehicle to be stuck and unable to get out. In order to overcome these problems, neural network and fuzzy logic approaches have been tried for this type of applications. Fuzzy logic approach [5, 6] seems promising, since it deals with various situations without requiring the construction of an analytical model of environment. While compared with the neural network approach [7], it has another distinct advantage that each rule of the rule base has a definite meaning and deals with a specific situation. This makes it possible to tune the rules manually. However, it is not easy to consistently construct the rules in the case of navigating the mobile vehicle in an unknown environment. To tackle this drawback, error back-propagation neural network was used to learn these rules[8]. Unfortunately, this method requires a sufficiently large set of representative patterns which can characterize the environment to train the network. To worsen the case, it is also difficult to obtain these training patterns which contain no contradictory input/output pairs. Thus reinforcement learning which requires only a scalar reinforcement signal as a performance feedback from the environment is quite attractive for constructing the fuzzy rule base. But due to the slow convergence speed of the reinforcement learning method, the Environment Exploration Methods (EEM), such as reference [9] which operates the mobile vehicle to explore a complex environment completely and have the rules constructed is time consuming and cannot guarantee to end up with sufficiently learned rules.

In this paper, we introduce a new navigation method which uses fuzzy logic and reinforcement learning, and propose a powerful method for constructing the fuzzy rule base. When compared with the EEM, it has four distinct advantages: (1) high learning speed; (2) high number of learned rule; (3) high adaptability (4) reliable convergence of learning network.

2 Overview of the navigator

As shown in Fig. 1, we assume that the vehicle is equipped with a ultrasonic sensor ring in which five sensor are equally distributed in space on the front semicircle. Each sensor, $s_i(i = 1,2,\cdots,5)$ gives the distance to the obstacle $d_i(i = 1,2,\ldots,5)$ in its field of view. Here we assume $4 \le d_i \le 200cm$ and each sensor covers a view of $\pi/4$. The control variables of the mobile vehicle are its linear velocity v and its steering angle $\Delta\theta$. In order to navigate the mobile vehicle to it's destination, we assume that the position vector of the mobile vehicle $p(X,Y)$ is always known from the internal sensor of the vehicle, and the goal vector $p_g(X,Y)$ is given. Therefore the navigation problem can be described as: given the input variables d_i, $p(X,Y)$ and $p_g(X,Y)$, control the output variables v and $\Delta\theta$ such that the vehicle avoids obstacles successfully and finally achieves its destination.

The structure of the proposed navigator is depicted in Fig. 2. The Navigation Supervisor, which is based on the If-Then rule, is a command module for controlling the mobile vehicle to move to its goal position or avoid obstacles. when the Obstacle Avoidor is activated, the Environment Evaluator performs an evaluation of the environment where the vehicle is, and determines the appropriate value of W for the Fuzzy Quantization. Further, input sensor readings are fuzzified and certain fuzzy inference is made. Finally the vehicle's action, v and $\Delta\theta$ are determined by defuzzifiction. In this paper we will focus on the Obstacle Avoidor and the method of constructing the fuzzy rule base.

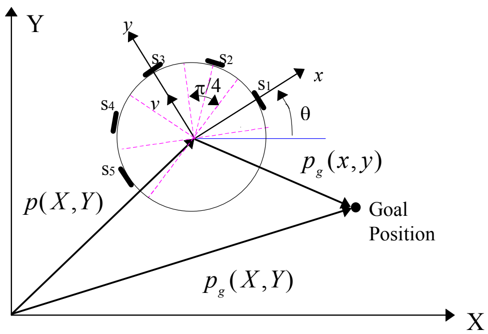

Fig. 1 Ultrasonic sensors and control variables

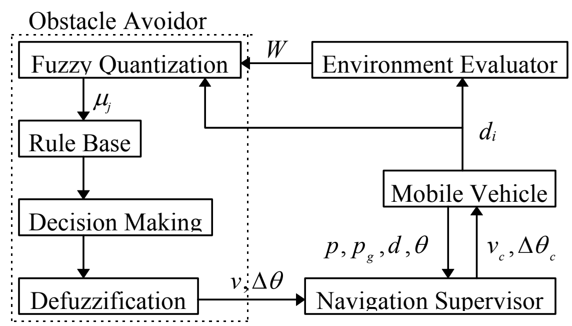

Fig. 2 Diagram of the proposed Navigator

3 Fuzzy control of obstacle avoidance

The Obstacle Avoidor is a fuzzy controller. The design steps of this module are as follows: (1) definition of membership functions for sensor input variables and control output variables; (2) fuzzification of the input variables; (3) rule base construction through reinforcement learning; (4) fuzzy inference; (5) defuzzification of the output variables.

Definition of the membership functions for the input/output variables and fuzzification of the input variables: The membership functions of the input and output variables are shown in Fig. 3. The crisp value of input variable d_i is fuzzified and expressed by the fuzzy sets - *VN*, *NR*, *FR*, which stand for very near, near and far, respectively. The fuzzy sets of the output variables v and $\Delta\theta$ have the definite membership functions, while their center positions (b_{1j} and b_{2j}, for $j = 1,2,\cdots,243$) are determined by the reinforcement learning.

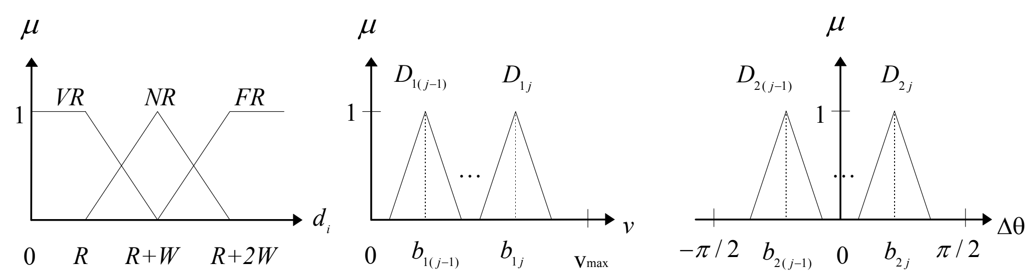

Fig. 3 The membership functions of input and output variables

Construction of the rule base and fuzzy reasoning: The fuzzy rules which play a role in mapping the sensor input space d_i to the mobile vehicle's action space v and $\Delta\theta$ are denoted by

Rule *j*: IF d_1 is D_{j1} AND ... AND d_5 is D_{j5} THEN v is V_j, $\Delta\theta$ is $\Delta\Theta_j$; for $j = 1,\cdots,243$

where $d_i (i = 1,2,\cdots,5)$ stands for the sensor inputs; $D_{ji}(i = 1,\cdots,5)$ are the fuzzy sets for d_i in the *jth* rule, which take the linguistic value of *VN*, *NR* or *FR*; v and $\Delta\theta$ denote the output variables; and V_j and $\Delta\Theta_j$ are the fuzzy sets for v and $\Delta\theta$. These rules are constructed through the reinforcement learning.

Let the fire strength of the *jth* rule be denoted by μ_j. For the inputs $d_1 = d_1', \cdots, d_5 = d_5'$, the fire strength of the *jth* rule can be written as

$$\mu_j = \mu_{D_{j1}}(d_1') \wedge \mu_{D_{j2}}(d_2') \wedge \mu_{D_{j3}}(d_3') \wedge \mu_{D_{j4}}(d_4') \wedge \mu_{D_{j5}}(d_5'); \quad \text{for } j = 1,\cdots,243 \tag{1}$$

If Mamdani's minimum operation is used for fuzzy implication, the memberships of the inferred fuzzy control action, V_j' and $\Delta\Theta_j'$ are calculated by

$$\mu_{V'}(v) = \bigcup_{j=1}^{243} \mu_j \wedge \mu_{V_j}(v) \quad \text{and} \quad \mu_{\Delta\Theta'}(\Delta\theta) = \bigcup_{j=1}^{243} \mu_j \wedge \mu_{\Delta\Theta_j}(\Delta\theta) \tag{2}$$

Defuzzification of output variables: For the reason of lower computing cost, we use the method of height defuzzification. The crisp control action is given by

$$v = \frac{\sum_{j=1}^{243} \mu_j b_{1j}}{\sum_{j=1}^{243} \mu_j} \quad \text{and} \quad \Delta\theta = \frac{\sum_{j=1}^{243} \mu_j b_{2j}}{\sum_{j=1}^{243} \mu_j} \tag{3}$$

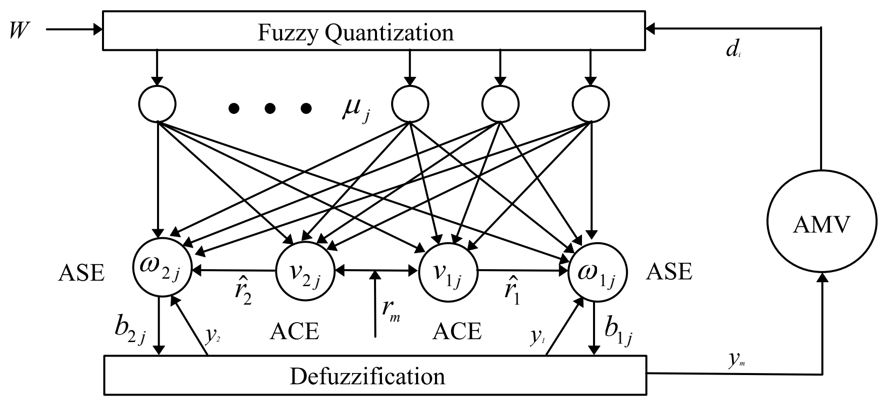

Fig. 4 Diagram of neural network to learn obstacle avoidance

4 Rule learning for obstacle avoidance

For convenience, we adopt the Sutton and Barto's model [10] in this paper. The structure of the learning algorithm is depicted in Fig. 4. The Fuzzy Quantization section encodes the input crisp value of sensors into μ_j by equation (1). In order to give the associativity in learning the rules, the trace, $\overline{\mu}_j(t)$ of the fired *jth* rule, is used. The trace at time step, *t+1* is given by

$$\overline{\mu}_j(t+1) = \lambda\overline{\mu}_j(t) + (1-\lambda)\mu_j(t) \tag{4}$$

where λ, $0 \le \lambda < 1$, is the trace decay rate. When a collision occurs, the associative critic element (ACE) receive an external reinforcement signal which is designed as

$$r_m = \begin{cases} -1 & \text{if } \min(d_i \mid i = 1,2,\cdots,5) < (R_{amv} + d_{s\min})(1+\varepsilon) \\ 0 & \text{otherwise} \end{cases} \quad \text{for } m = 1,2 \tag{5}$$

where R_{amv} is the radius of the mobile vehicle, $d_{s\min}$ is the minimum distance which the ultrasonic sensor can detect, and ε, $0 < \varepsilon < 1$, is a safety factor. While using the temporal difference learning theory, the prediction of the external reinforcement signal is

$$p_m(t) = E\left(\sum_{t' \ge t} \gamma^{t'-t} r_m(t'+1)\right) \tag{6}$$

where $0<\gamma<1$, and $E(\bullet)$ denotes the expectation. Thus if $p_m(t)$ is correctly learned, then

$$p_m(t-1) = r_m(t) + \gamma p_m(t) \tag{7}$$

In the case of incorrect learning, an internal reinforcement signal will be generate by equation (7) to train the ACE, which is defined as

$$\hat{r}_m(t) = r_m(t) + \gamma p_m(t) - p_m(t-1) \tag{8}$$

The prediction value, $p_m(t)$ is implemented as follows:

$$p_m(t) = G\left(\sum_{j=1}^{243} v_{mj}(t)\mu_j(t)\right) \tag{9}$$

Where $G(x) = 2/\left(1+e^{-\xi x}\right)-1$. In order to predict $p_m(t)$ correctly, the weights of ACE must be updated. It is expressed by

$$v_{mj}(t+1) = v_{mj}(t) + \beta\hat{r}_m(t)\bar{\mu}_j(t) \tag{10}$$

where β is a positive constant determining the rate of change of v_{mj}. In the same way, the weights of the associative search element (ASE) are updated by

$$\omega_{mj}(t+1) = \omega_{mj}(t) + \alpha\hat{r}_m(t)e_{mj}(t) \tag{11}$$

where α, $0 < \alpha \le 1$, determines the learning rate, $e_{mj}(t)$ is the eligibility of the *jth* rule at time *t*, which is updated by

$$e_{mj}(t+1) = \delta e_{mj}(t) + (1-\delta)y_m(t)\mu_j(t) \tag{12}$$

where δ, $0 \le \delta < 1$, is a trace decay rate, and $y_m(t) = (v, \Delta\theta)^T$ is the control action at time step t. Eventually, if the rules are sufficiently learnt in a specific environment, the weights of ASE will converge to fixed values. The center position of the fuzzy sets (Fig. 4) at each time step are determined by

$$b_{mj}(t) = b_m + \frac{\omega_{mj}(t)f_m}{k\max\left(\left|\omega_{mj}(t)\right|\right) + \left|\omega_{mj}(t)\right|} \tag{13}$$

where b_m, k and f_m are constants. When the learning process is terminated, the learned $b_{mj}(t)$ determined by equation (13) at the time step are used as the collision avoidance rule base for the mobile vehicle in its real navigation.

5 Simulation and results analysis

As we mentioned above, the reinforcement learning methods that have been proposed in the literature typically converge slowly. This drawback limits their capability on solving simple learning task. The EEM could be used to learn the rules for obstacle avoidance. But it cannot avoid the disadvantages that: (1) it is time consuming in exploring the environment; (2) whether the rules are sufficiently learnt or not cannot be ascertained when terminating the learning process; (3) it is not clear that an optimum environment for training the vehicle exists.

In this paper, we proposed a method to tackle this problem. Our key idea is to separate the rule learning and real navigation into two stages: (1) in the rule learning stage, we use a small value of *W* (Fig. 5), so the rule learning process could be implemented in a very small and simple environment; (2) in the real navigation stage, the Environment Evaluator is activated to generate an appropriate parameter-*W* of the current environment where the vehicle is. As a result, the fuzzy rule base learned in a small environment can be adaptively used in a new and fully unknown environment.

$R = 28cm$	$\delta = 0.85$	$\lambda = 0.5$	$b_1 = 15cm/s$	$b_2 = 0$	$\beta = 0.8$	$\alpha = 0.8$
$W = 20cm$	$\gamma = 0.95$	$\varepsilon = 0.2$	$f_1 = 15cm/s$	$f_2 = \pi/2$	$k = 0.2$	$\xi = 1.5$

Table 1. Parameter Used for simulation

A simple corridor-like environment (Fig. 5) was used to train the vehicle. Since the environment is regular, the trajectory of the vehicle has been kept unchanged when the learning method converges. The parameters for the simulation are shown in table 1. At the beginning of the simulation, $v_{mj}(t)$ was set to small non-zero values, while $\omega_{mj}(t)$, $\overline{\mu}_j(t)$, $p_m(t-1)$, $e_{mj}(t)$ were set to zero. The mobile vehicle began its first trial of the learning steps, which consists of a series of learning steps until a collision occurs. When a collision occurred, it backtracked 4 steps and the heading direction was reversed, then the next trial began. After several collisions, the vehicle navigated successfully in counter-clockwise (CCW) direction (which direction the vehicle takes in the environment depends on the start point.) and kept a constant trajectory. Then the CCW training was terminated and the clockwise (CW) direction was repeat. If a collision occurs, the vehicle will backtrack 40 steps and the heading angle turn π/30 in the CW direction, then the next trial began. After the vehicle navigated successfully in the CW direction and maintained its trajectory unchanged, the whole learning process was completed. The performance of the learning process could be expressed as the number of steps which was taken before a collision occurred. Simulation was carried out to compare our method with the EEM. In the EEM, the vehicle was trained in the environment shown in Fig. 8 with the same parameters as our method. The rules were almost sufficiently learnt (1.2% of the rules were blank) in our method after 34 trials, where CCW took up 22 trials since it began with a blank rule base, and CW took up 12 trials. On the other hand, the EEM took 100 trials and the rule base was far from sufficient (30% of the rules were blank). It only handled 3000 learning steps before a new collision occurred. Further simulation showed that the rule base was not constructed sufficiently at up to 40000 learning steps. Fig. 6 shows that the proposed method is quite efficient than EEM. In some situations where the EEM caused the vehicle to go into a dead loop, the performance of learning deteriorated significantly.

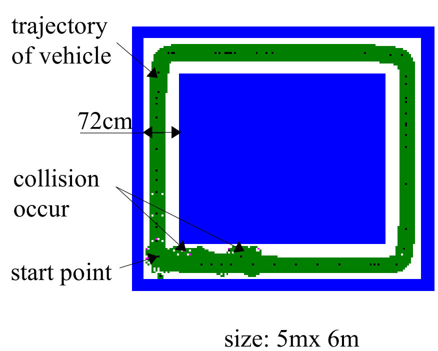

Fig. 5 Learning in a simple environment

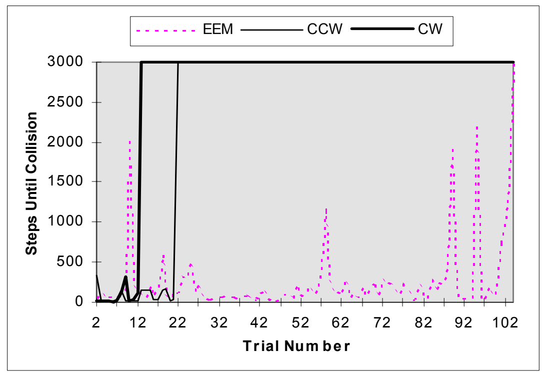

Fig. 6 Performance of the proposed training method

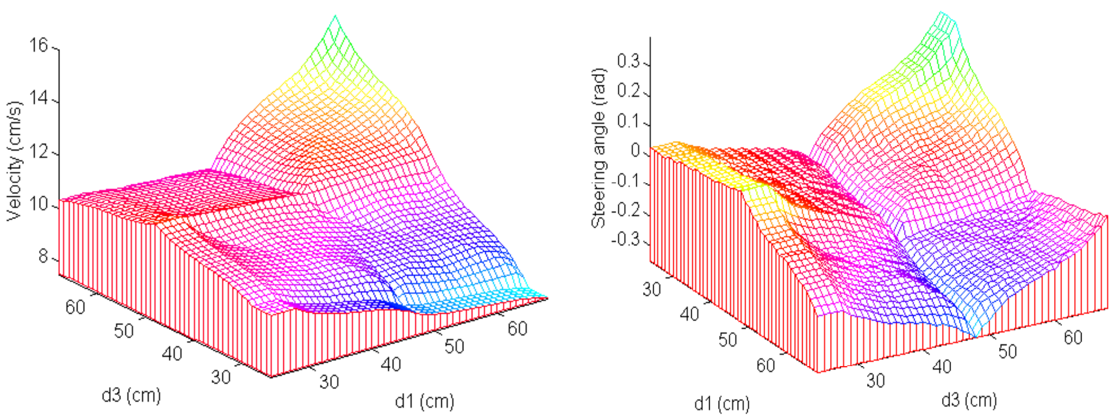

Fig. 7 Control surfaces of the collision avoidor

In the case of $W=20$cm, we found that the optimum width of the corridor for training the vehicle was 72cm. It means that we could get the largest number of rules in this case. For instance, there were only 3 blank rules for the width of 72cm while there were 18 blank rules for the width of 80cm. If the width is larger, it becomes more difficult to train the vehicle, as it turns in a zigzag trajectory along the corridor. For this reason, the rules learnt in the width of 72cm are used as the obstacle avoidance rule base. Furthermore, the three blank rules could be constructed in two ways: (1) re-train the vehicle in the situation associated with these rules until it learns to handle with the situation. (2) add the three rules manually. For convenience, the second method was adopted.

After the construction of the rule base, we plotted the control surfaces of the rule base. Whether the rules are learnt correctly and sufficiently could be shown by these control surfaces. Fig. 7 shows an example of $d_2=d_4=d_5=38cm$, it is indicated in this figure that if d_3 is near and d_1 is far, the vehicle will turn a negative Δθ (in the CW direction). This action enables it to move into a collision-free region. In addition, it is found that the velocity is reduced while d_1 or d_3 become small. The adaptability of the rule base was tested in an unknown environment (Fig. 8) by simulation. The vehicle navigated 300000 steps, completely explored the environment without a single collision. However, since the rules are learnt with a small W. this makes the obstacle avoidor rather nearsighted. So the Environment Evaluator was used in the proposed navigator to generate an appropriate W in the simulated navigation. The function of the Environment Evaluator is that it determines a large W in a environment of low obstacle density to ensure the vehicle "sees" farther and determines a small W in a environment with high obstacle density to ensure it is capable to navigate through this environment. The impact of the evaluator is that it reduces the average velocity (Fig. 9). The whole navigator is tested by simulation as shown in Fig. 8.

6 Conclusion

We have proposed an efficient method to learn the fuzzy rule base of collision avoidance. Based on this method, we have further proposed a new scheme for an intelligent navigator, in which an Environment Evaluator was introduced to tune the universe of discourse of the input variables. The learning algorithm and the scheme for the whole navigator have been verified and proved to be successful by a series of simulation. In our future research, The proposed method will find its application in multi-behavior navigation of mobile vehicle in a dynamic environment.

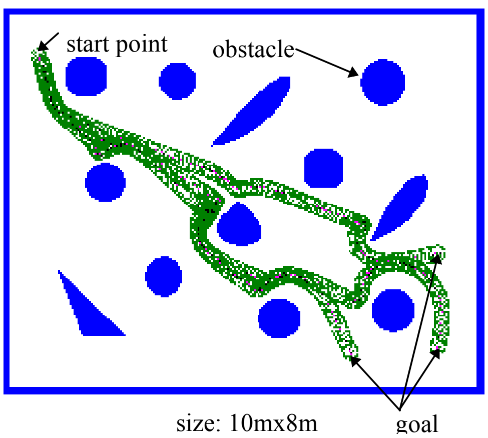

Fig. 8 Navigation in a complex environment

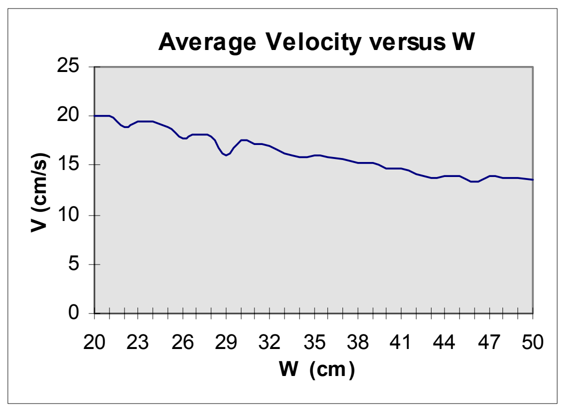

Fig. 9 Impact of tuning W

References

[1] Rohnert, H., "Shortest paths in the plane with convex polygonal obstacles", Information Processing Letters, 23,71-76,1986.

[2] Lozano-Perez, T., "Spatial planning: A configuration space approach", IEEE Trans. on Computers, C-32(2), 108-120, 1983.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", Int. J. of Robotics Research, vol. 5, no. 1, pp. 90-98, 1986.

[4] J. Borestein and Y. Koren, "Real-time obstacles avoidance for fast mobile robot", IEEE Trans. Syst. Man Cyber., vol. 19, no. 5, pp. 1179-1187, Sep./Oct. 1989.

[5] K. T. Song and J. C. Tai, "Fuzzy navigation of a mobile robot", Proc. of IEEE/RSJ International conference on Intelligent Robotics and Systems, pp. 621-627, 1992.

[6] F. G. Pin and H. Watanabe, "Using custom-designed VLSI fuzzy inferencing chips for the autonomous navigation of a mobile robot", Proc. of IEEE/RSJ International conference on Intelligent Robots and Systems, pp. 790-795, 1992.

[7] Prabir K. Pal and Asim Kar, "Mobile robot navigation using a neural net", Proc. of IEEE International Conference on Robotics and Automation, pp 1503-1508, 1995.

[8] C. Kozakiewwicz and M. Ejiri, "Neural network approach to path planning for two dimension robot motion", Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 818-823, 1991.

[9] H. R. Beom and H. S. Cho, "A sensor-based navigation for a mobile robot using fuzzy Logic and Reinforcement Learning", IEEE Trans. Syst. Man Cyber., vol. 25, No. 3, pp. 464-477, March, 1995.

[10] Barto , R. S. Sutton, and C. W. Anderson, "neuronlike adaptive elements that can solve difficult learning control problems", IEEE Trans. Syst. Man Cyber., vol. SMC-13, no. 5, Sep./Oct.,1983.