Self-learning Fuzzy Navigation of Mobile Vehicle

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 describes a self-learning navigation method which utilizes fuzzy logic and reinforcement learning for navigation of a mobile vehicle in uncertain environments. The proposed navigator consists of three modules: Obstacle Avoidance, Move to Goal and Fuzzy Behavior Supervisor. The fuzzy rules of the on-line obstacle avoidance are learnt through reinforcement learning. A new and powerful method is proposed to constructed these rules automatically. The effectiveness of the learning method and the whole navigator are verified by simulation.

1. Introduction

Path planning is an important issue in navigation of mobile vehicles, which could be classified into three general approaches: roadmap [1], cell decomposition [2] and potential field [3,4]. The first two are carried out in an off-line manner in completely known environments. They are not suitable for navigation in complex, unknown or dynamic environments. The potential field method seems quite efficient in on-line path planning, however, it has two disadvantages: (1) It is difficult to find the force coefficients influencing the velocity and direction of the mobile vehicle in a clutter environment which are too complex to be embedded in a mathematical model; (2) The potential local minimum could cause the vehicle to be stuck there and unable to get out. In this situation, fuzzy logic approach seems promising [5,6], however, it is not easy to consistently construct the rules in the case of navigating the mobile vehicle in a clutter environment. To tackle this problem, reinforcement learning has been introduced to learning these rules automatically [7], but due to the slow learning rate, methods such as the Environment Exploration Method (EEM) [7,8] which operates the mobile vehicle to explore the environment and has the rules constructed, is time consuming and cannot guarantee to end up with a sufficient number of learned rules.

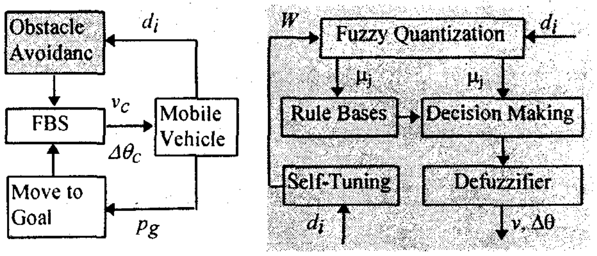
In this paper we describe a new navigation method which uses fuzzy logic and reinforcement learning, and propose a powerful method for constructing the fuzzy rule base. It has four advantages: (1) high learning speed, (2) high number of learned rules, (3) high adaptability, and (4) reliable convergence of learning network.

2. Navigator overview

The navigation system is depicted in Fig. 1 (a). It consists of three modules. Both the Obstacle Avoidance and Move to Goal behaviors are based on fuzzy control, their rule bases are constructed through reinforcement learning. The Fuzzy Behavior Supervisor (FBS) is a command module to fuse the two behaviors and generate the correct motion command for the vehicle. In this paper we will focus on the Obstacle Avoidance module and the construction of the fuzzy rule base.

3. Algorithm of collision-free navigation

The Obstacle Avoidance module is shown in Fig. 1 (b). With the appropriate W generated by the self-tuning module, the input sensor readings is fuzzified and fuzzy inference is made. Finally the vehicle's action: the linear velocity and it's steering angle, are correctly determined by defuzzification. The design of this module is as follows: (1) fuzzification of the input variables, (2) rule base construction through reinforcement learning, (3) defuzzification of the output variables.



(a) (b)

Fig. 1 Diagram of the proposed Navigator

3.1 Sensor reading and control variables

Assuming the vehicle is equipped with five ultrasonic sensors equally and spatially distributed on the front semicircle on the x-y plane, each of which gives the distance to the obstacle $d_i (i = 1,\cdots,5)$ in its field of view.

0-7803-2912-0

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 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 $p_g(X,Y)$, controls the output variables v and $\Delta\theta$ such that the vehicle avoids obstacles successfully and finally achieves its destination.

3.2 Fuzzy control of the obstacle avoidance

As shown in Fig. 2, 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 their definite membership functions whose center positions (b_{1j} and b_{2j}) will be determined by reinforcement learning. The fuzzy rule base which plays a role in mapping 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 $\Delta\Theta_j$ $(j=1,\cdots,243)$

where d_i stands for sensor input; D_{ji} is the fuzzy set for d_i in the *jth* rule, which is one of *VN*, *NR*, *FR*; v and $\Delta\theta$ denotes the output variables; V_j and $\Delta\Theta_j$ are the fuzzy sets for v and $\Delta\theta$. Let the fire strength of *jth* rule be denoted by μ_j. For the inputs $d_1 = d_1^{'},\cdots,d_5 = d_5^{'}$, the fire strength of *jth* rule can be denoted by

$$\mu_j = \mu_{D_{j1}}(d_1^{'}) \wedge \mu_{D_{j2}}(d_2^{'}) \wedge \cdots \wedge \mu_{D_{j5}}(d_5^{'}) \qquad (1)$$

If Mamdani's minimum operation is used for the fuzzy implication and the height defuzzification method is used for defuzzification, the crisp control action is given by

$$v = \frac{\sum_1^{243} \mu_j b_{1j}}{\sum_1^{243} \mu_j} \quad \text{and} \quad \Delta\theta = \frac{\sum_1^{243} \mu_j b_{2j}}{\sum_1^{243} \mu_j} \qquad (2)$$

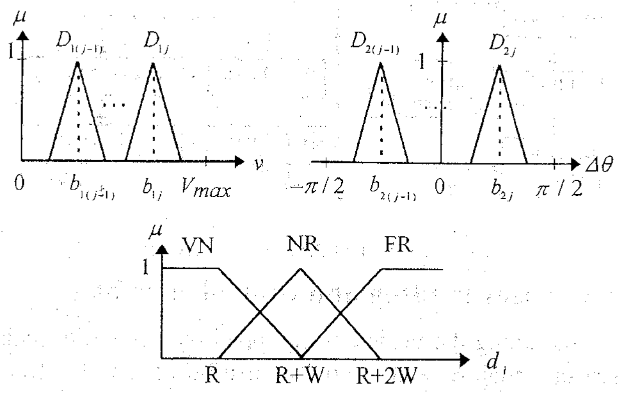


Fig. 2 The membership functions of input and out put variables

3.3 Rule learning for obstacle avoidance

The approaches which utilize reinforcement learning to learn obstacle avoidance have been introduced by [7][8]. Ref. [7] is based on fuzzy control and [8] is not. However, both of them are based on EEM. These methods at least have three disadvantages: (1) It is time consuming to explore the environment; (2) whether the rules are sufficiently learned or not cannot be ascertained upon the termination of the learning process; (3) the number of learned rules cannot be certain. Furthermore, it is not clear that an optimum environment for training the vehicle exists. In this paper, we proposed a method to tackle this problem. Since we have no intention to proposed a fully new reinforcement learning theory, we adopt Sutton and Barto's model [9], and give a brief description of the learning method.

The structure of learning method is depicted in Fig. 3. The Fuzzy Quantization encodes the input crisp value of sensors into μ_j by equation (1).

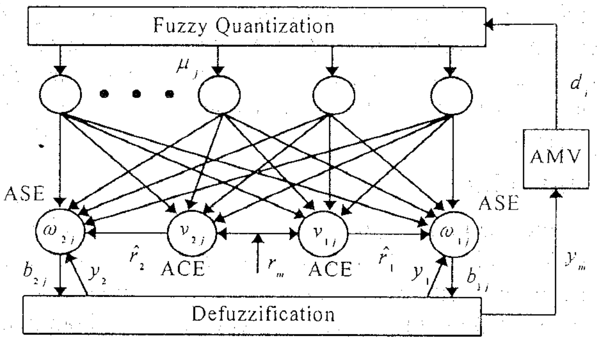


Fig. 3 Diagram of neural network to learn obstacle avoidance

In order to give the associativity in learning the rules, the trace, $\overline{\mu}_j(t)$ of the fire *jth* rule is used. The trace at time step, $t+1$ is given by

$$\overline{\mu}_j(t+1) = \lambda\overline{\mu}_j(\mathrm{t}) + (1-\lambda)\mu_j(\mathrm{t}) \qquad (3)$$

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

$$r_m = \begin{cases} -1 & \text{if } \min(d_i) < (R_{amv} + d_{s\min})(1+\varepsilon) \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

where $m=1,2$, 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

$$p_m(t) = E\left(\sum_{t^{'} \ge t} \gamma^{t^{'}-t} r_m(t^{'}+1)\right) \qquad (5)$$

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

$$p_m(t-1) = r_m(t) + \gamma p_m(t) \qquad (6)$$

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

$$\hat{r}_m(t) = r_m(t) + \gamma p_m(t) - p_m(t-1) \qquad (7)$$

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

$$p_m(t) = G\left(\sum_{j=1}^{243} v_{mj}(t)\mu_j(t)\right) \qquad (8)$$

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

$$v_{mj}(t+1) = v_{mj}(t) + \beta\hat{r}_m(t)\overline{\mu}_j(t) \qquad (9)$$

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

$$\omega_{mj}(t+1) = \omega_{mj}(t) + \alpha\hat{r}_m(t)e_{mj}(t) \qquad (10)$$

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

$$e_{mj}(t+1) = \delta e_{mj}(t) + (1-\delta)y_m(t)\mu_j(t) \qquad (11)$$

where δ, $0\leq\delta<1$, is a trace decay rate, and $y_m(t) = (v, \Delta\theta)^T$ is the control actions at time step t. Eventually, if the rules are sufficiently learnt in a specific environment, the weights of ASE will converge to a fixed values. The center position of the fuzzy sets (Fig. 3) at each time step are determined by

$$b_{mj}(t) = b_m + \frac{\omega_{mj}(t)f_m}{k\max\left(\left|\omega_{mj}(t)\right|\right) + \left|\omega_{mj}(t)\right|} \qquad (12)$$

where b_m, k and f_m are constant. While the learning process terminates, the learned $b_{mj}(t)$ at the time step is used as the collision avoidance rule base for the mobile vehicle in real navigation.

3.4 Implementation of rule learning

In this paper, we separate the rule learning and real navigation into two stages: (1) In the rule learning stage, we use a small value of W (Fig. 2), and implement learning in a very small and simple environment; (2) In the real navigation stage, the self-tuning module is activated to tune the universe of discourse of the sensor input by generating an appropriate value of W for 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 fully unkown environment.

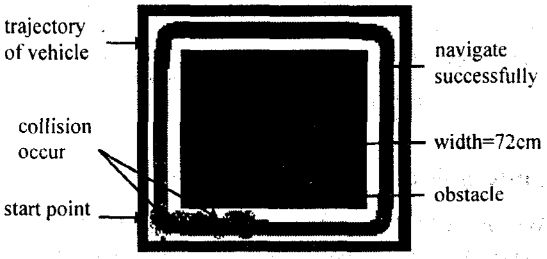
4. Simulation and results

In the simulation, we used a corridor-like environment (Fig. 4) to learn the fuzzy rules. Since the environment is regular, when the learning algorithm converges, the trajectory of the vehicle will be kept unchanged, thus the learning process could be terminated with a sufficient number of learned rules. The parameters of the simulation are as follows: (1) the distance which the ultrasonic sensors can detect is from 4cm to 200cm; (2) R and W are set to be 28cm and 20cm respectively; (3) The other parameters are shown in table 1.

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

Table 1. Parameter Used for simulation

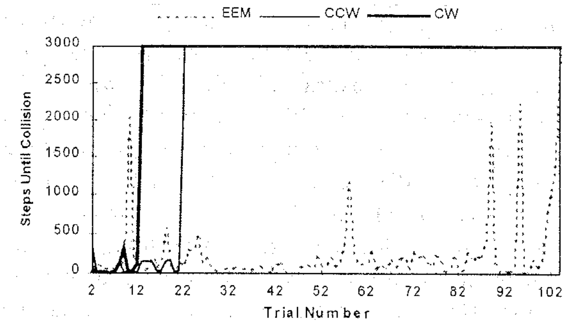
For W=20cm, The optimum width of the corridor for learning is 72cm. At the beginning of the simulation, $\omega_{mj}(t)$ is set to small nonzero values, while $v_{mj}(t)$, $\overline{\mu}_j(t)$, $p_m(t-1)$, $e_{mj}(t)$ are set to zero. The mobile vehicle begins its first trial of the learning steps which consists of a series of learning steps until a collision occurs. When a collision occurs, it will backtrack 4 time steps and the heading direction will be reversed, and a new trial begins. The performance of the learning process can be expressed as the number of steps that were taken before a collision occurred. The vehicle was trained in the clockwise (CW) and counter-clockwise (CCW) directions, until the trajectory of both of these directions converged, meaning the weights of the ASE have achieved their fixed values. Simulation was carried out to compare our method with the EEM. In the EEM, the vehicle was trained in the environment shown in Fig. 7 with the same parameters as our method. The rules are sufficiently learned (it can be verified by the control surface of the controller) in our method after 34 trials, where CCW takes up 22 trials since it begins with a blank rule base, and CW takes up 12 trials. However the EEM takes 100 trials and the rule base is far from sufficient (30% of the rules is blank). It only handled 3000 learning steps before a new collision occurred. Further simulation shows that the rule base is not constructed sufficiently at 40000 learning steps. Fig. 5 shows that the proposed method is quite efficient as compared with the EEM. In some situations when the EEM caused the vehicle to go into a dead loop, the performance of learning become worse.



size: 5mx6m

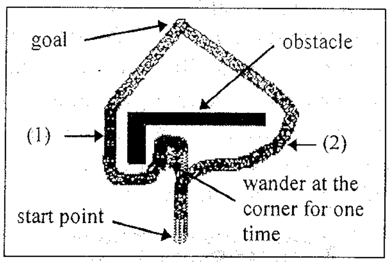
Fig. 4 Learning in a small and simple environment

After the construction of the rule bases, we test its adaptability in the environment as shown in Fig. 7 by simulation. The vehicle navigated 200000 steps, explored the whole environment without collision. However, since the fuzzy control is based on a small W, and the output of the control algorithm is keep unchanged while the distance of the obstacle is lager than $R+2W$, it has two disadvantages: First, the obstacle avoidance algorithm is very nearsighted. This resulted in that the vehicle might take an unreasonable trajectory (Fig. 6) in real navigation. Second, the change of v and $\Delta\theta$ is large in one time interval. In order to avoid these disadvantages, we use a self-tuning module to determine an appropriate W for the Obstacle Avoidance module in real navigation. This make the vehicle "see" farther (Fig. 6) and smoothly move around obstacles. However, it reduces the average velocity of the vehicle. The whole navigator is tested by simulation as shown in Fig. 7. In addition, the navigator is free of local minimum (Fig. 8)



EEM - Trained in the environment shown in Fig. 7
CW, CCW - Trained in the environment shown in Fig. 4

Fig. 5 Performance of the proposed training method

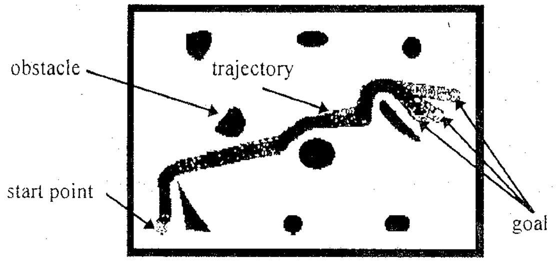


(1) Trajectory without self-tuning (2) Trajectory with self-tuning

Fig. 6 Testing the self-tuning module

5. Conclusion

We proposed an efficient method to learn the fuzzy rule base of collision avoidance. Based on this method, we proposed a new scheme of an intelligent navigator, in which a self-tuning mechanism was used to tune the universe of discourse of the input variables. The proposed method will find its application in multi-behavior navigation of mobile vehicle in our future research.



size: 10mx8m

Fig. 7 Navigation from a start point to three destinations

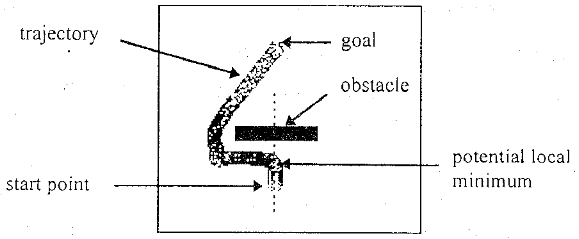


Fig. 8 Navigation without local minimum

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 Tras. 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 for fast mobile robot", IEEE Trans. Syst. Man Cyber., vol. 19, no. 5, pp. 1179-1187, Sep./Oct. 1989.
5. 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. 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.
8. A. G.Ben J.A. Krose and Joris W.M. van Dam, "Adaptive state space quantization for reinforcement learning of collision-free navigation", Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1327-1332, 1992.
9. 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.