ELSEVIER

Fuzzy Sets and Systems 87 (1997) 133–140

FUZZY sets and systems

A neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments [1]

Wei Li[a,*], Chenyu Ma[a], F.M. Wahl[b]

[a] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, People's Republic of China*

[b] *Institute for Robotics and Computer Control, Technical University of Braunschweig, Hamburger Str. 267, W-3300 Braunschweig, Germany*

Received April 1995; revised December 1995

Abstract

A neuro-fuzzy system architecture for behavior-based control of a mobile robot in unknown environments is presented. A neural network is used to understand environments. Its inputs are a heading angle between the robot and a specified target, and range information acquired by an array of ultrasonic sensors. The output from the neural network is a trained reference motion direction for robot navigation.

The methodology of the behavior-based control approach proposed in this paper is: (1) to analyze and to decompose a complex task based on stimulus–response behavior; and (2) to quantitatively formulate each type of behavior with a simple feature by fuzzy sets and fuzzy rules as well as to coordinate conflicts and competition among multiple types of behavior by fuzzy reasoning. An advantage is that building fuzzy sets and rules for each simple-featured type of behavior is much easier than for a complex task.

Based upon a reference motion direction and distances between the robot and obstacles, different types of behavior are fused by fuzzy logic to control the velocities of the two rear wheels of the robot. Simulation experiments show that the proposed neuro-fuzzy system can improve navigation performance in complex and unknown environments. In addition, this architecture is suitable for multisensor fusion and integration. © 1997 Elsevier Science B.V.

Keywords: Robotics; Engineering; Artificial intelligence; Sensor-based motion planning; Behavior-based control

1. Introduction

In robot applications in the real world, a mobile robot should be able to operate in uncertain and dynamic environments. Behavior-based control [1, 2] shows potentialities for robot navigation in unknown environments since it does not need building an exact world model and complex reasoning process. Before, however, behavior-based control is used to navigate a mobile robot in the real world perfectly, much effort should be made to solve problems with it, such as, the quantitative formulation of behavior, and the efficient coordination of conflicts and competition among multiple types of behavior, and so forth. In order to overcome these deficiencies, some fuzzy-logic-based behavior control schemes have been proposed [4, 5, 9, 10].

* Corresponding author.

[1] This research was supported by the National Science Foundation of China under grant 69585004 and by the Chinese "863" High Technology Project under grant 863-512-28-20.

PII S0165-0114(95)00015-2

Based upon fuzzy logic, speed control and turn control of a mobile robot are determined by goal orientation and obstacle proximity in [9]. The idea of this approach starts with an implementation of very low level actions using very few rules. However, some types of behavior with a high level, such as, *following edges*, are not realized, so navigation performance could deteriorate when the mobile robot needs the behavior, *following edges*, to escape from a room without global information. The methodologies of the strategies in [5, 10] are similar. That is: first, the idea of stimulus–response behavior is used to analyze and to decompose a complex task; then, fuzzy rules are used to formulate each type of behavior with simple feature as well as fuzzy reasoning is used to fuse different types of behavior. The differences between [5] and [10] lie in concrete implementation, such as, type selection of behavior and control schemes.

In comparison with traditional approaches [1, 2], the fuzzy-logic-based approach fuses different types of behavior using fuzzy reasoning [13] rather than simply inhibiting some types of behavior according to an assigned priority. Consequently, unstable oscillations between different types of behavior can be avoided. This approach also differs from fuzzy control approaches for obstacle avoidance in [8, 11, 12] since perception and decision units in this method are integrated in one module based on reactive behaviors, and they are directly oriented to a dynamic environment to improve real-time response and reliability.

In the control scheme proposed in [5], the input signals are the distances between the robot and the obstacles to the left, front, and right locations as well as the heading angle between the robot and a specified target. In analogy to artificial potential fields [3], the distances between the robot and obstacles serve as a repulsive force for avoiding obstacles, while the heading angle serves as an attractive force for moving to the target. But the fuzzy-logic-based approach is orthogonal to strict geometrical computation on environments, so it is more robust than the artificial potential field approach. Since, however, this control scheme always uses the heading angle as a reference motion direction to do behavior fusion, it differs, to some extent, from the way a human would drive a car, so that it does not guarantee to provide a good path for robot navigation in some cases. The graphical simulation in

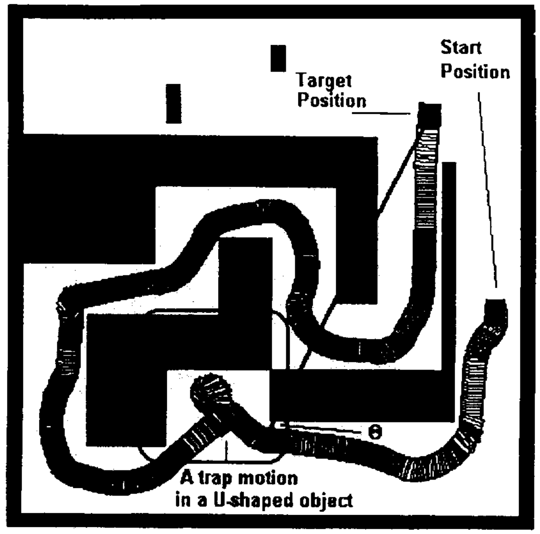

Fig. 1. Robot navigation with a trap motion in a U-shaped object caused by a heading angle.

Fig. 1 shows that the robot has a trap motion in a U-shaped object since it uses the heading angle as its reference motion direction during its whole navigation. In fact, a driver only takes the heading angle as a reference motion direction to drive his car, when there are no obstacles between his car and a target; otherwise the driver must determine a reference motion direction according to the distribution of obstacles in local regions.

In order to improve navigation performance in unknown environments, this paper presents a new neuro-fuzzy system architecture [7] by adding a neural network into the control scheme in [5]. The inputs to the neural network are the heading angle between the robot and a specified target, and range information acquired by an array of ultrasonic sensors, and the output from the neural network is a trained reference direction for robot motion. Based upon the reference motion direction and distances between the robot and obstacles in dynamic environments, behavior fusion is done by fuzzy logic to control the velocities of the two rear wheels of the robot. Simulation experiments show that the proposed neuro-fuzzy system can improve navigation performance in complex and unknown environments. In addition, this architecture is suitable for robot navigation by multisensor fusion and integration [6].

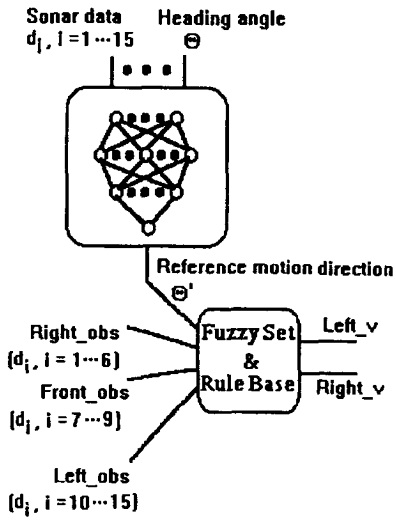

Fig. 2. A neuro-fuzzy system architecture for mobile robot navigation in uncertain environments.

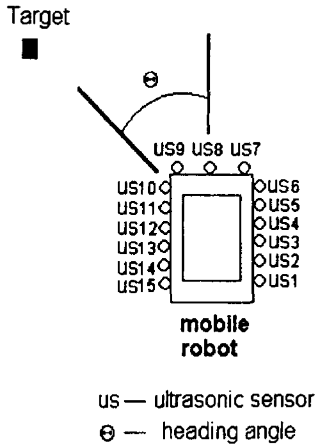

Fig. 3. Ultrasonic sensors for a mobile robot navigation.

2. Neuro-fuzzy system architecture for robot navigation

Fig. 2 shows the proposed neuro-fuzzy system architecture for robot navigation in unknown environments. The conception of the navigation system architecture is based on the combination of a neural network and the fuzzy logic navigation scheme proposed in [7].

The neural network is used to process range information to determine a good reference motion direction. Therefore, its input signals are the heading angle, Θ, between the robot and a specified target, and the sonar data, d_i ($i = 1, \ldots, 15$), that are acquired by 15 ultrasonic sensors, mounted on the THMR-II mobile robot with 1.0 m length and 0.8 m width, as shown in Fig. 3. These ultrasonic sensors are divided into three groups to detect obstacles to the left, front, and right locations, respectively. When the target is located to the left side of the mobile robot, the heading angle, Θ, is defined as negative; when the target is located to the right side of the mobile robot, the heading angle, Θ, is defined as positive. The output from the neural network is a reference motion direction, Θ', for robot navigation.

In the fuzzy control scheme, fuzzy rules and fuzzy reasoning are used to formulate all types of behavior quantitatively and to coordinate their conflicts and competition. The input signals to the fuzzy control scheme are distances between the robot and obstacles to the left, front, and right locations, denoted by left_obs, front_obs, and right_obs, respectively, as well as a reference motion direction, Θ', which is determined by the neural network. The outputs from the fuzzy control system are the results of behavior fusion to control the speed of the two rear wheels of the mobile robot. The linguistic variables far, med(medium) and near are chosen to fuzzify left_obs, front_obs, and right_obs. The linguistic variables P(positive), Z(zero), and N(negative) are used to fuzzify the reference motion direction, Θ'; the linguistic variables fast, med, and slow are used to fuzzify the velocities of the driving wheels left_v and right_v. Fig. 4 shows their corresponding membership functions.

Here, it is noted that the sonar data d_i ($i = 1, \ldots, 15$) have been independently processed by both the neural network and the fuzzy control scheme for different purposes, respectively. For the neural network, the sonar data d_i ($i = 1, \ldots, 15$) in combination with the heading angle, Θ, are used as its input patterns to train a good motion direction. In the fuzzy control scheme, the sonar data, d_i ($i = 1, \ldots, 15$), are used for computing the distances between the robot and obstacles, right_obs, front_obs, and left_obs, in the real world as follows:

$$\text{right\_obs} = \text{Min}\{d_i\}, \quad i = 1, \ldots, 6, \qquad (1)$$

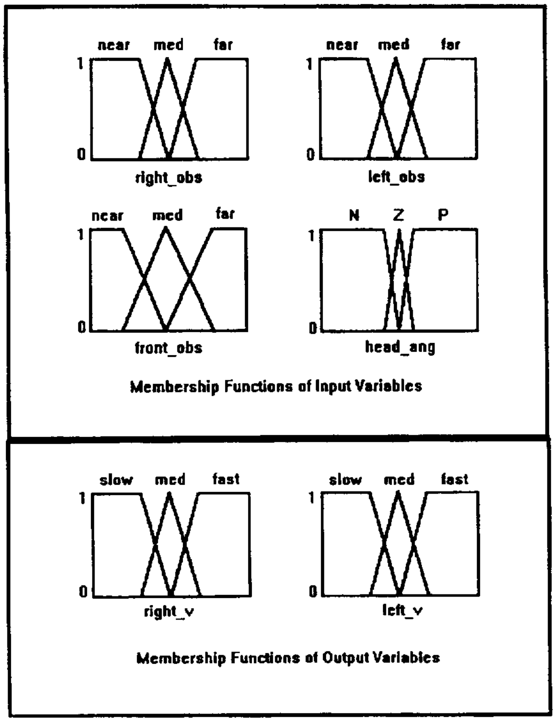

Fig. 4. Membership functions regarding input and output variables.

$$\text{front\_obs} = \text{Min}\{d_i\}, \quad i = 7, \ldots, 9, \tag{2}$$

$$\text{left\_obs} = \text{Min}\{d_i\}, \quad i = 10, \ldots, 15. \tag{3}$$

The strategy for independently processing sonar data d_i ($i = 1, \ldots, 15$), using both the neural network and the fuzzy control scheme, can drastically reduce the effects of sonar data errors on navigation performance. This is because: (1) The neural network only needs the relative relations of sonar data d_i rather than the absolute precise values of sonar data d_i to train reference motion directions. (2) The fuzzy control system only needs the minimum values, right_obs, front_obs, and left_obs, derived from the sonar data d_i ($i = 1, \ldots, 15$), to avoid obstacles. Consequently, even though one of the processing sonar data d_i causes some errors, another is able to compensate such errors to some extent. If the neural network outputs a *false* reference motion direction, this reference does not cause robot collision with obstacles since a real motion direction of the robot is not only determined by its reference motion direction, but also by distances between the robot and obstacles,

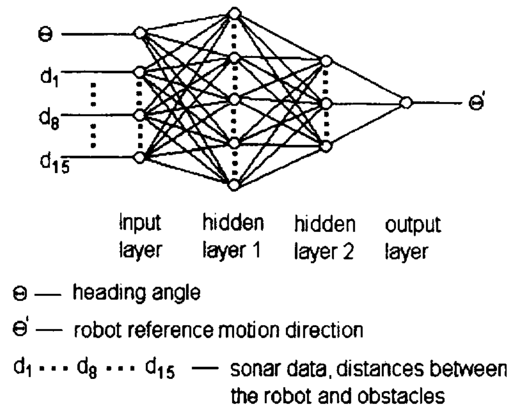

Fig. 5. The used neural network for training robot reference motion direction.

right_obs, front_obs, and left_obs. For example, the graphical simulation in Fig. 1 shows that the robot reaches the target without collision with obstacles, although its reference motion directions are identical with its heading angles during the whole navigation. In another case, even if some of right_obs, front_obs, and left_obs have errors, the possibility of robot collision with obstacles can be reduced when the neural network outputs a *good* reference motion direction.

3. Training reference motion directions by neural network

In the neuro-fuzzy system architecture, a four-layer standard back-propagation (BP) network is used to train robot reference motion directions, as shown in Fig. 5. The inputs to the neural network are the sonar data d_i ($i = 1, \ldots, 15$), representing the distribution of obstacles in local regions, and the heading angle between the robot and a specified target. Fig. 6 lists some of the circumstance patterns. The following examples are used to explain how such circumstance patterns can be used for training robot reference motion directions.

1. Fig. 6(a) shows that a reference motion direction, Θ', is identical with the heading angle, Θ, since there are no obstacles around the robot ($d_i = d_{\max}$; $i = 1, \ldots, 15$).

2. Fig. 6(b) shows that a reference motion direction, Θ', is zero since there exist obstacles on the left

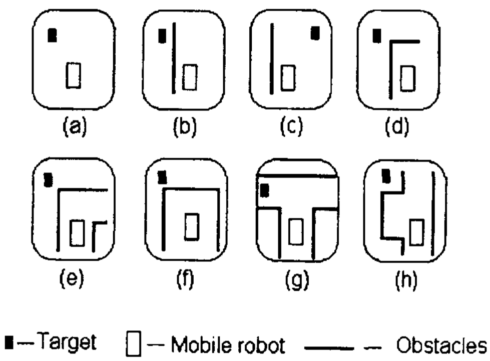

Fig. 6. Circumstance patterns for neural network.

side of the robot ($d_{\min} < d_i < d_{\max}$; $i = 10, \ldots, 15$, and $d_i = d_{\max}$; $i = 1, \ldots, 9$) although the heading angle, Θ, is negative.

3. Fig. 6(c) shows that a reference motion direction, Θ', is identical with the heading angle, Θ, since there are no obstacles between the robot current position and the target although there exist obstacles on the left side of the robot ($d_{\min} < d_i < d_{\max}$; $i = 10, \ldots, 15$, and $d_i = d_{\max}$; $i = 1, \ldots, 9$). ...

According to such circumstance patterns, the input–output relationship of each unit can be written as follows.

Input units:

$$q_i^{[1]} = d_i, \quad i = 1, \ldots, 15, \qquad q_{16}^{[1]} = \Theta. \tag{4}$$

Hidden units:

$$q_j^{[s]} = f(\mathrm{Net}_j^{[s]}), \quad s = 2, 3, \tag{5}$$

$$\mathrm{Net}_j^{[s]} = \sum_i w_{ji}^{[s]} \cdot q_i^{[s-1]}, \quad s = 2, 3. \tag{6}$$

Output unit:

$$\hat{\Theta}' = q^{[4]} = f(\mathrm{Net}^{[4]}), \tag{7}$$

$$\mathrm{Net}^{[4]} = \sum_i w_i^{[4]} \cdot q_i^{[3]}, \tag{8}$$

where $w_{ji}^{[s]}$ is the weight on connection joining the ith neuron in layer $[s-1]$ to the jth neuron in layer $[s]$, and $f(x)$ is a sigmoid logistic function:

$$f(x) = \frac{1}{1 + \mathrm{e}^{-x}}. \tag{9}$$

The error, $E = 0.5 \sum_k (\Theta'_k - \hat{\Theta}'_k)^2$, is used to modify the weight $w_{ji}^{[s]}$ by the following δ learning rule:

Output layer:

$$\delta^{[4]} = f'(\mathrm{Net}^{[4]}) \sum_k (\Theta'_k - \hat{\Theta}'_k). \tag{10}$$

Other layers:

$$\delta_j^{[s]} = f'(\mathrm{Net}_j^{[s]}) \sum_k (\delta_k^{[s+1]} \cdot w_{kj}^{[s+1]}), \quad s = 1, 2, 3. \tag{11}$$

The connection weights of the neural network are updated by

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t+1), \tag{12}$$

$$\Delta w_{ij}(t+1) = \eta \delta_j^{[s]} \mathrm{Net}_i^{[s-1]} + \alpha \Delta w_{ij}(t), \tag{13}$$

where η is a learning coefficient and α is a momentum constant.

4. Behavior fusion by fuzzy reasoning

A key issue of behavior-based control is how to efficiently coordinate conflicts and competition among different types of behavior to achieve a good performance. A usual approach to coordinating multiple types of behavior is to fire a behavior according to an inhibiting and suppressing strategy associated with artificial potential fields [2]. The following are some deficiencies of this strategy noted in our experiments:

1. In some cases, the robot cannot reach a given target. Fig. 7 shows that the robot is unable to get through the narrow channel to reach the given target. The reason is that the robot always activates *obstacle avoidance* behavior when it approaches this channel, so that it turns to the right to move into a large free space.

2. Much effort must be made to test and to adjust some thresholds for firing each behavior during preprogramming. Especially, these thresholds depend heavily on environments, i.e., a set of thresholds, determined in a given environment, may not be suitable for others.

3. Robot motion with unstable oscillations between different types of behavior may occur in some cases. This is because just only one type of behavior could be

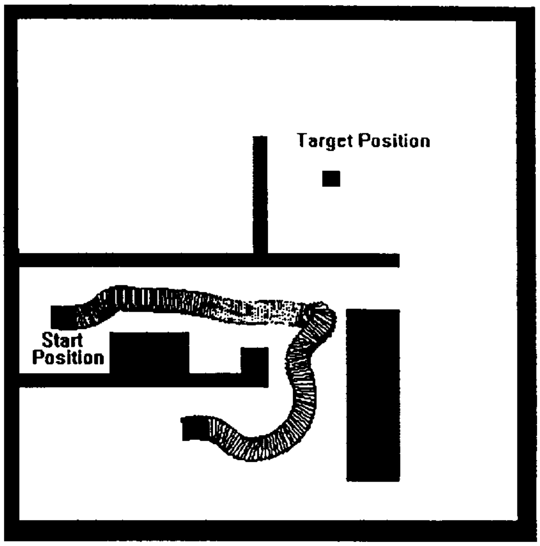

Fig. 7. Robot navigation by priority strategy.

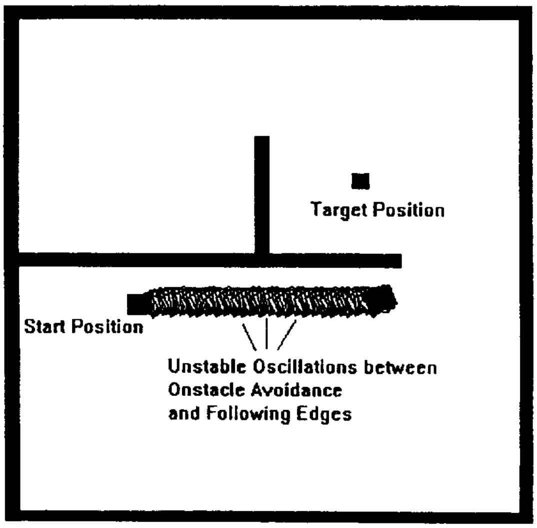

Fig. 8. Unstable oscillations caused by behavior control according to priority strategy.

activated at a given instant and two types of behavior with neighboring priority, e.g., *obstacle avoidance* and *following edges*, are fired in turn, as shown in Fig. 8.

In order to do behavior fusion by fuzzy logic, first of all, fuzzy sets and fuzzy rules are used to formulate each type of behavior, such as (1) obstacle avoidance; (2) following edges; (3) target steer; (4) decelerating at curved and narrow roads [5]. Then, all types of be-

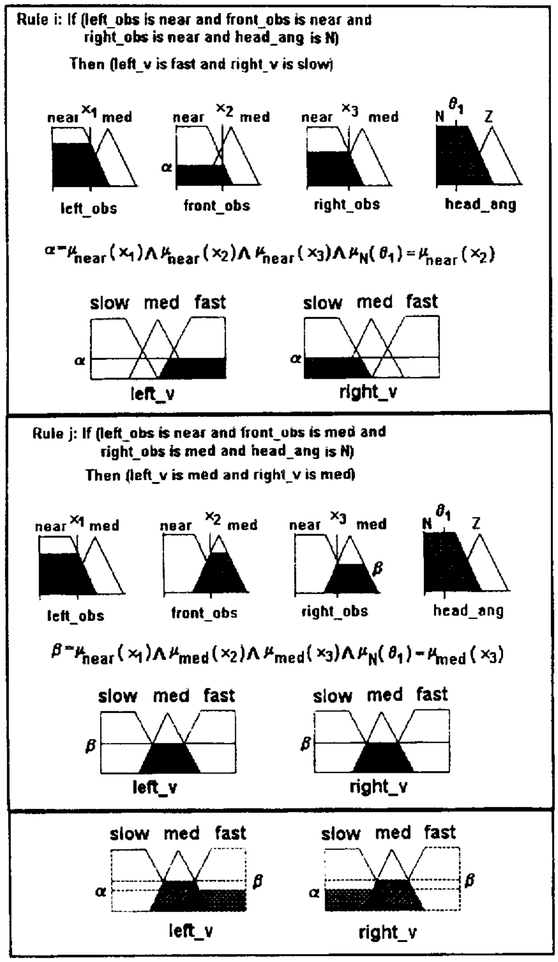

Fig. 9. Behavior fusion by fuzzy reasoning.

havior are weighted by fuzzy reasoning. The following is an illustration of how this problem is dealt with by the Min–Max inference algorithm and the centroid defuzzification method, as shown in Fig. 9. For instance, the inputs left_obs $= x_1$, front_obs $= x_2$, and right_obs $= x_3$, $\Theta' = \theta_1$, are fuzzified by their membership functions to fire fuzzy rules associated with them simultaneously. Assume that *Rule i* (see below), formulating the *obstacle avoidance* behavior, and *Rule j* (see below), formulating the *following edges* behavior, are fired according to the fuzzified inputs (in fact, much more fuzzy rules may be activated):

Rule i: *If* (left_obs *is* near *and* front_obs *is* near and right_obs *is* near *and* head_ang *is* N)
Then (left_v *is* fast *and* right_v *is* slow).

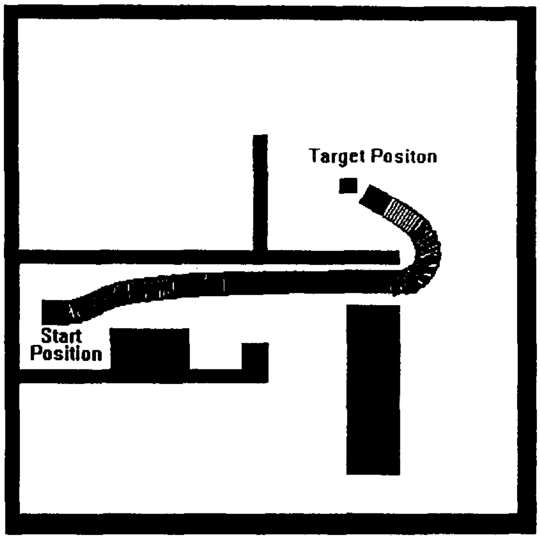

Fig. 10. Robot navigation by behavior fusion.

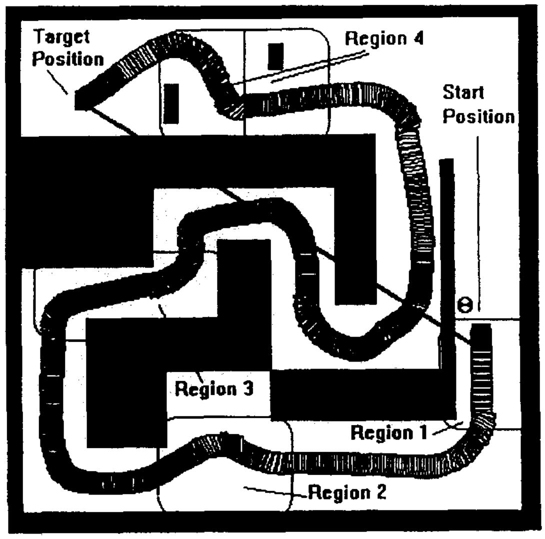

Fig. 11. Robot navigation by the neuro-fuzzy system.

Rule j: *If* (left_obs *is* near *and* front_obs *is* med *and* right_obs *is* med *and* head_ang *is* N) *Then* (left_v *is* med *and* right_v *is* med).

By fuzzy reasoning, both *Rule i* and *Rule j*, related to the *obstacle avoidance* and *following edges* behaviors respectively, are weighted to determine an appropriate control action, i.e., the velocities, left_v and right_v, of the robot's rear wheels. By using behavior fusion based on fuzzy reasoning, robot navigation performance can be greatly improved. Fig. 10 shows that the robot reaches the target in Fig. 7 by efficiently weighting multiple types of behavior, such as avoiding obstacles, following edges, and moving toward target and so on.

5. Simulations

To demonstrate the effectiveness of the proposed neuro-fuzzy system, a simulation experiment on robot navigation in an unknown environment is shown in Fig. 11. In local region 1, the robot moves according to its reference motion direction, $\Theta' = 0$, outputted by the neural network, although the heading angle, Θ, in region 1 is positive. In local region 2, using the neural network the robot recognizes that there is a U-shaped object around it, and outputs a *good* reference motion direction to avoid such trap motion in Fig. 1. In local region 3, the robot gets through a narrow and curved road by multi-behavior fusion and local reference motion directions. It can be observed that the robot automatically reduces its speed to avoid collision when it moves in such a narrow and curved road. In local region 4, the robot first uses the heading angle as its reference motion direction because it cannot detect obstacles in its front; the robot then turns right to avoid an obstacle by using a positive reference motion direction when it detects its obstacle. When the robot goes past the obstacle, it reaches the target by using the heading angle.

6. Conclusions

In this paper, a new neuro-fuzzy system architecture for behavior-based control of robot navigation in uncertain environments is proposed. This strategy consists of two levels: (1) the high level is for environment understanding; and (2) the low level is for behavior control. At the high level, a neural network is used to process range information for understanding the distribution of obstacles in local regions; while at the low level fuzzy sets and fuzzy rules are used to formulate each type of behavior quantitatively and to coordinate conflicts and competition among multiple types of behavior efficiently. In general, the more information on environments are obtained at the high level, the more correct decision can be made at the low level. Especially, the strategy for independently processing sonar data, using both the neural network and the fuzzy control scheme, can drastically reduce the effects of sonar data errors on navigation performance. The simulation results demonstrate that, using this system, navigation performance in complex and unknown environments can be greatly improved.

Reference

[1] R.C. Arkin and R.R. Murphy, Autonomous navigation in a manufacturing environment, *IEEE Trans. Robotics Automation* **6** (1990) 445–454.

[2] R.A. Brooks, A robust layered control system for a mobile robot, *IEEE J. Robotics Automation* **2** (1986) 14–23.

[3] O. Khatib, Real-time obstacle avoidance for manipulators and automobile robots, *Internat. J. Robotics Res.* **5** (1986) 90–98.

[4] W. Li, Robot obstacle avoidance and navigation using fuzzy logic in uncertain environments, Technical Report, Department of Computer Science and Technology, Tsinghua University (1993).

[5] W. Li, Fuzzy logic based "perception-action" behavior control of a mobile robot in uncertain environments, *Proc. 1994 IEEE Conf. on Fuzzy Systems*, Orlando, FL (26–29 June 1994) 1626–1631.

[6] W. Li, Fuzzy logic based robot navigation in uncertain environments by multisensor integration, *Proc. 1994 IEEE Internat. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, Las Vegas, Nevada, USA (2–5 October 1994) 259–264.

[7] W. Li, Neuro fuzzy systems for intelligent robot navigation and control under uncertainty, *Proc. 1995 IEEE Conf. on Fuzzy Systems*, Yokohama, Japan (20–24 March 1995) 1747–1754.

[8] M. Maeda, Y. Maeda and S. Murakami, Fuzzy drive control of an autonomous mobile robot, *Fuzzy Sets and Systems* **39** (1991) 195–204.

[9] F.G. Pin and Y. Watanabe, Steps toward sensor-based vehicle navigation in outdoor environments using fuzzy behaviorist approach, *J. Intelligent Fuzzy Systems* **1** (1993) 95–107.

[10] A. Saffiotti, E.H. Ruspini and K. Konolige, Blending reactivity and goal-directedness in a fuzzy controller, *Proc. 1993 IEEE Conf. on Fuzzy Systems*, San Diego, CA (8–12 March 1993) 134–139.

[11] M. Sugeno and M. Nishida, Fuzzy control of model car, *Fuzzy Sets and Systems* **16** (1985) 103–113.

[12] T. Takeuchi, Y. Nagai and N. Enomoto, Fuzzy control of a mobile robot for obstacle avoidance, *Inform. Sci.* **45** (1988) 231–248.

[13] L.A. Zadeh, Fuzzy sets, *Inform. and Control* **8** (1965) 338–353.