Multi-Behavior-Based Path Planning for Indoor Mobile Robot

Hongxu Chen
College of Computer Science and Technology
Chongqing University of Posts and Telecommunications
Chongqing, China
chenghongxu777@163.com

Guotian He
Chongqing Institute of Green and Intelligent Technology
Chinese Academy of Sciences
Chongqing, China

***Abstract*—To figure out a local path planning for mobile robot during its autonomous navigation indoors, the paper presents a behavioral fusion path planning on the basis of fuzzy logic, in which positioning method of robot is based on RFID technology. Through using twenty-four ultrasonic sensors to collect information in the external environment, it integrates behavioral control with fuzzy control to create a path planner made up of Goal Seeker, Obstacle Avoider, Obstacle Following Behavior Controller, Behavior Weighting Controller, and finally brings up a new fusion behavioral algorithm, whose validity and feasibility are confirmed by tests in its software emulation platform and physical robot system.**

Keywords- indoor mobile robot; path planning; fuzzy logic; behavior fusion

I. Introduction

The application of indoor mobile robot is becoming more and more extensive in people's production and life, which can be used as narrator in exhibition hall, shopping guide in mall, receptionist, guide in hospital and other places.

Path planning is the core of mobile robot navigation. Path planning of mobile robot means a robot can find out a path from the starting point to the destination in a work environment with obstacles so that the robot can bypass all obstacles and reach the destination without collision [1];

At present, the robot path planning at home and abroad includes artificial potential field, A^* algorithm, grid method and other mature approaches. As the method of artificial potential field is easy to fall into local minimum, the robot cannot reach the goal position[2]; the method of A^* algorithm should not only consider the searching frequency of a search node, but also the number of nodes with a great lot of calculation and high complexity[3]; for grid method, grid is used to represent environment, but there is conflict between resolution of environment and the storage of environmental information, the huge calculation limits its use scope[4]; it is very difficult to establish the precise mathematical model in the movement of mobile robot due to the uncertainty of complex environment and dynamic change, in contrast, the fuzzy logic control algorithm embodies great advantage, which does not require to establish mathematical model for the control system. The mobile robot is a typical nonlinear, time-delayed and unstable system, and the fuzzy controller can complete the nonlinear mapping from the input space to the output space; and fuzzy logic method is able to combine the robustness of the fuzzy control with "perception - action" behavior [5-7] based on physiology, and proposes new ideas for navigation of mobile robot in complex and uncertain environment.

II. The Path Planning Based on Behavior

The path planning tactic based on behavior is to divide the robot navigation into several independent simple act units. Each unit is a motion control unit with corresponding navigation function made of actuator and sensor. Each unit that takes independent operation mode has a corresponding sub-goal, and these units complete the task of navigation through mutual cooperation with each other [8-9].

In the path planning based on behavior, it will be very convenient and efficient to establish each behavior module with fuzzy logic, so we often combine the behavior control with fuzzy logic control; the information detected by the sensor can be used as the input variable of each module's behavior after fuzzy processing; design the basic behavior by designing subset of fuzzy rules and membership functions contained in the fuzzy subset; different behavior selects different parameters to control the movement of the robot, and the fusion of fuzzy behavior is to use suitable operator to normalize the fuzzy output of each behavior, then obtain the actual precise control amount of the robot with the method of defuzzification[10].

III. Robot System Model

The mobile robots used in this experiment is mainly composed of the three-tier architectures; the bottom is track chassis drive system, including a drive unit composed by a DC motor; the middle layer is a sensor layer with 24 ultrasonic sensors and RFID readers; the upper is for data processing and control decision-making, mainly consisting a laptop and robot control software mounted therein.

A. Sensor layout

In the robot system used in this experiment, the 24 ultrasonic transducers are uniformly distributed on the circle with radius of R_V=25cm; the sensor layout is shown in Fig. 1. Sensor No.i is S_i (i=1,2, …,24); the Y-axis shows the forward direction of the robot; the sensor on the positive part of X-axis is S_2, and then the number of sensors is recursive counterclockwise. The sensor located on the positive part of Y-axis is sensor S_8, and the obstacles distance detected by sensor i is l_i (4cm≤li≤400cm). The angular range detected by each sensor is 15 degrees. To better detect and avoid obstacles, the 15 sensors S_1~S_{15} in the front of the robot are divided into five sensor groups sg_i(i=1,2,3,4,5), and each sensor group sg_i composes of 3 adjacent sensors S_{3I-2}, S_{3I-1}, S_{3I}, the 5 sensor groups cover an angular range of 225 degrees in the front.

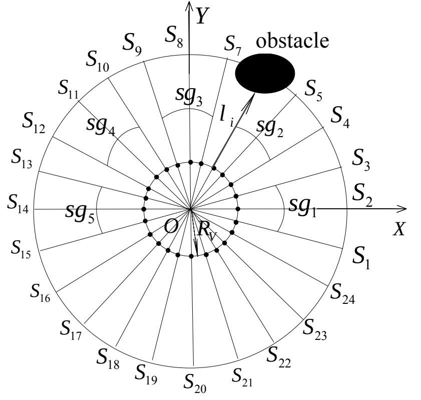


Figure 1. The sensor layout in the robot

Within the range of each sensor group sg_i, the shortest distance from the robot center to the obstacle we detected is defined as d_i (i=1,2,3,4,5), and

$$\delta_\iota = P_\varsigma + \mu\iota\nu\ (\lambda_\varphi | \varphi = 3\iota-2, 3\iota-1, 3\iota) \quad (1)$$

To form the sensors into group has two advantages:

(1) Reduce the dimension of input, and reduce the complexity of the navigation methods. (2) Reduce the possibility of false detection of obstacles.

B. Coordinate system and control variables

Two coordinate systems need to be built throughout the navigation process of robots, one is the world coordinate system XOY, the other is the robot coordinate system xoy, as shown in Fig. 2.

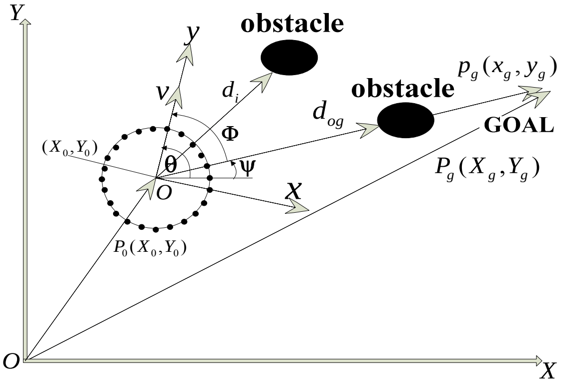


Figure 2. Coordinate system and control variables schematic

In the above figure, XOY represents the world coordinate system, and xoy represents the robot coordinate system. v represents the movement direction of the robot. θ represents lead angle of the robot (the included angle between the movement direction of the robot and the positive x axis of the world coordinate). ψ indicates the included angle between the goal direction and the positive x-axis. Φ represents the included angle between the goal direction and the positive y axle. d_{og} represents the obstacle distance along the direction of the goal. P_O ($X_{O,}$ Y_O) represents the coordinate of robot center in the world coordinate system. P_g (X_g, Y_g) represents the coordinate of the goal point in the world coordinate system. p_g (x_g, y_g) represents coordinate of the goal point in the robot coordinate system.

C. Navigation and position system

In the entire navigation process of mobile robot, it is one of the necessary conditions of path planning that the moving robot can make precise positioning by itself. C219156A type 433MHz active RFID reader and C129071A type 433MHz active RFID tag are used to establish the positioning system in this experiment.

(1) Wireless Signal Strength Ranging Principle

RFID system uses the received signal strength indication (RSSI) to indicate RF signal strength in a certain position. In general, in the active RFID system, the signal strength emitted by each label is a fixed value; the signal decays in the propagation process, when the distance between the tag and readers is different. The signal strength value received by the reader is different, so we use the theoretical and empirical propagation loss model [11]. The distance between the tag and reader can be calibrated in the use of the strength of the received signal. The correspondence between the signal strength value and distance is shown in the following formula:

$$PL(d) = PL(d_0) + 10n\lg(\frac{d}{d_0}) + x_\varepsilon \quad (2)$$

According to the above transmission loss model, we can get the distance d from the tag to reader, and the function equation among the tag signals strength received by reader:

$$d = 10^{[P(d_0)-P(d)]/10n} \quad (3)$$

(2) Positioning System and Positioning Algorithm

In this experiment, the RFID reader is installed in a central location of the robot, and the placed arrangement of RFID tag is shown in Fig. 3. If the label is placed on the floor, it might be artificially trampled and crushed by robot track, so it is arranged on the ceiling. In Fig. 3, the labels are divided into grids with axes. The separation distance between labels is 1.5m, the coordinate values where each label located is written in the tag. When the robot equipped with a reader moves in the area, the robot can read multiple tags. It selects 4 nearest tags to read for their coordinate values and RSSI values, the distance between the robot and 4 nearest labels can be calculated based on four RSSI values by using equation (2). The equation set formed by 4 equations can be obtained in the use of the 4 coordinate values and 4 distance values. The coordinate value at the current position of the robot can be obtained after this equation is solved, as well as the precise positioning of the robot. As shown in Fig4.

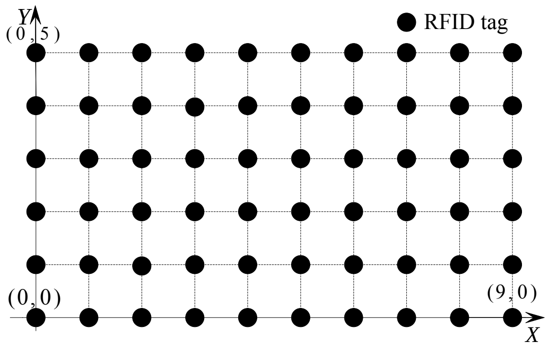


Figure 3. Schematic diagram of RFID tag arrangement

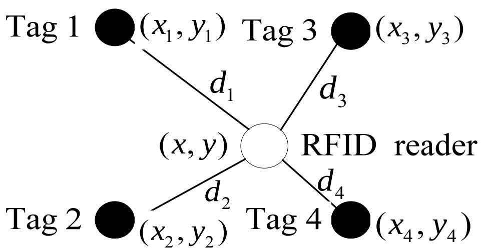


Figure 4. Positioning method

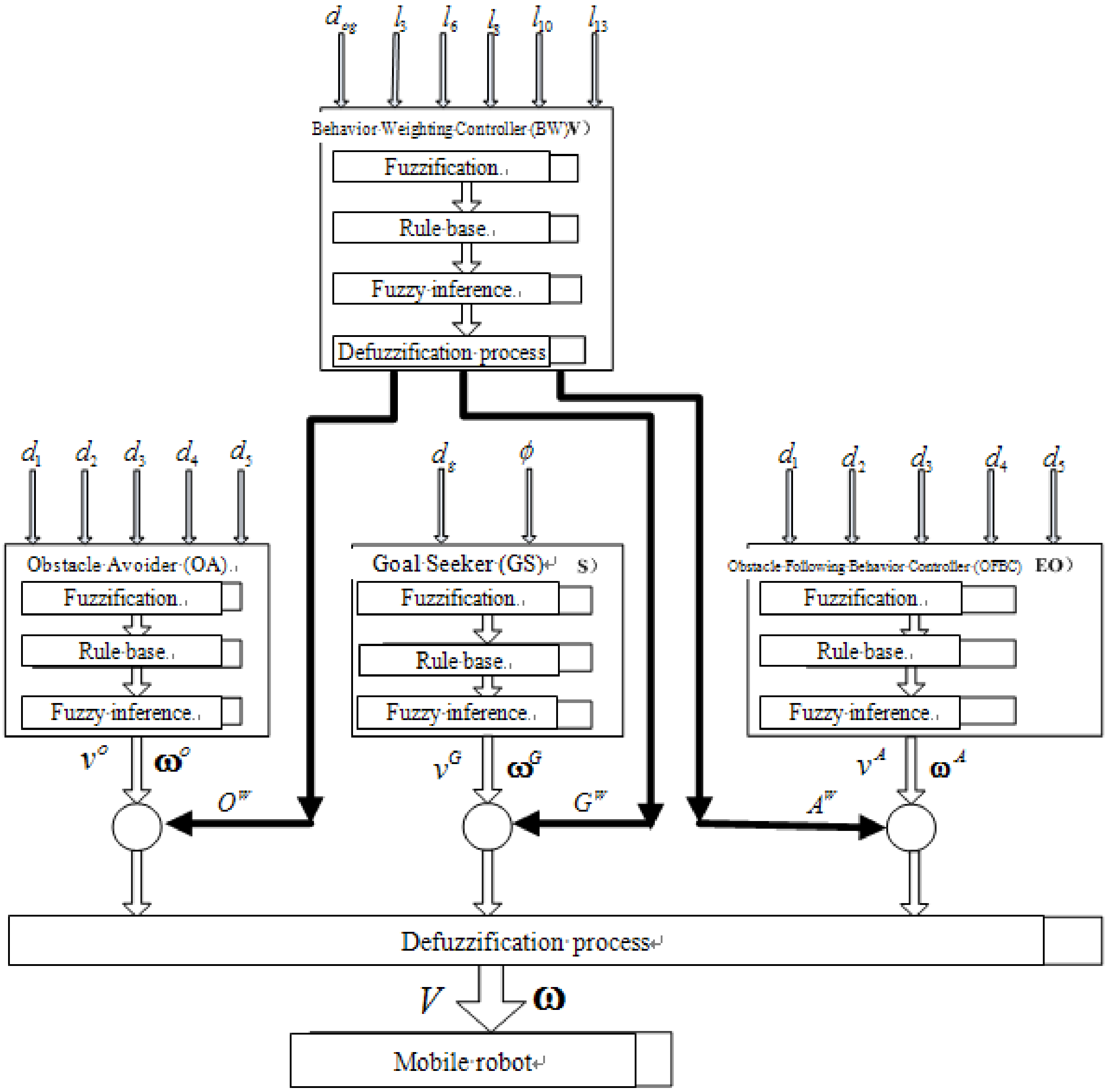


Figure 5. Behavioral path planner based on fuzzy logic.

IV. The Path Planner

The path planner mainly consists of four parts: Obstacle Avoider (OA), Goal Seeker (GS), Obstacle Following Behavior Controller (OFBC), and Behavior Weighting Controller (BW), as shown in Fig. 5. These 4 controllers are based on fuzzy logic.

A. Goal seeker

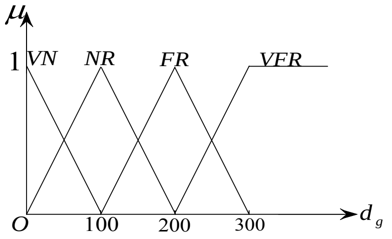
Single goal seeking behavior does not consider the obstacle, terrain and other factors. It can ensure the robot reaches the goal position from the starting position. The input variables of the fuzzy controller are Φ and d_g, wherein

$$d_g = \sqrt{(X_g - X_0)^2 + (Y_g - Y_0)^2} \tag{4}$$

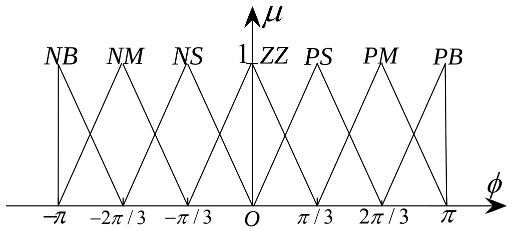
$$\phi = \theta - \psi \tag{5}$$

$$\psi = \tan^{-1}\left(\frac{y_g}{x_g}\right) \tag{6}$$

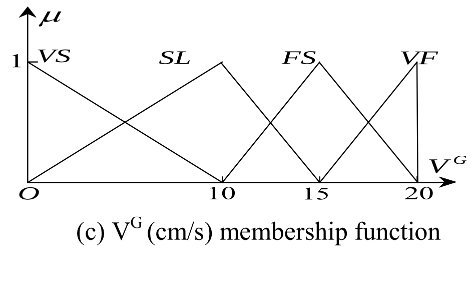
Output variables include line speed of the robot v^G and angle change ω^G. The membership functions of the four variables are shown in Fig. 6.



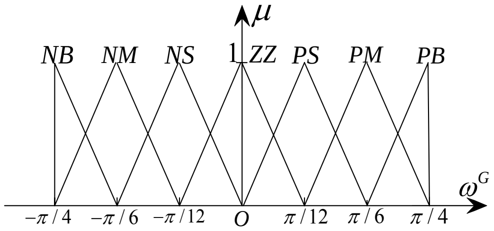
(a) d_g (cm) membership function



(b) ϕ (rad) membership function



(c) V^G (cm/s) membership function



(d) ω^G (rad) membership function

Figure 6. The fuzzification of goal seeking behavior

As shown in Fig. 6, the input variable d_g is fuzzified into a fuzzy set with four fuzzy languages {VN, NR, FR, VFR}(VN: very near, NR: near, FR: far, VFR: very far). The input variable Φ is fuzzified into a fuzzy set with seven fuzzy languages {NB, NM, NS, ZZ, PS, PM, PB}(NB: negative big, NM: negative middle, NS: negative small, ZZ: zero, PS: positive small, PM: positive middle, PB: positive big). Output variable V^G is fuzzified into a fuzzy set with four fuzzy language {VS, SL, FS, VF}(VS: very slow, SL: slow, FS: fast, VF: Very fast). Output variable ω^G is fuzzified into a fuzzy set with seven languages {NB, NM, NS, ZZ, PS, PM, PB}. Since this controller has 2 input variables, the fuzzy subsets are respectively 4 and 7, so there is a total of 28 rules, as shown in Table 1.

TABLE 1. The goal seeking behavior fuzzy rules table

Φ / d_g	NB	NM	NS	ZZ	PS	PM	PB
VN	(VS, PB)	(VS, PM)	(VS, PS)	(VS, ZZ)	(VS, NS)	(VS, NM)	(VS, NB)
NR	(VS, PB)	(SL, PM)	(SL, PS)	(SL, ZZ)	(SL, NS)	(SL, NM)	(VS, NB)
FR	(SL, PB)	(SL, PM)	(FS, PS)	(FS, ZZ)	(FS, NS)	(FS, NM)	(SL, NB)
VFR	(SL, PB)	(FS, PM)	(VF, PS)	(VF, ZZ)	(VF, NS)	(SL, NM)	(SL, NB)
Controlling behavior (V^G , ω^G)							

If the input of fuzzy variable is $\phi = \phi'$, $d_g = d_g'$, then the confidence coefficient of each rule among the 28 rules

$$\mu_j(\phi', d_g') = \mu_{\Phi_j}(\phi') \wedge \mu_{D_j}(d_g') \text{ (φ=1,2,□,28)}, \tag{7}$$

If Mamdani inference rule is used, the membership functions of the fuzzy output v^G and ω^G are respectively

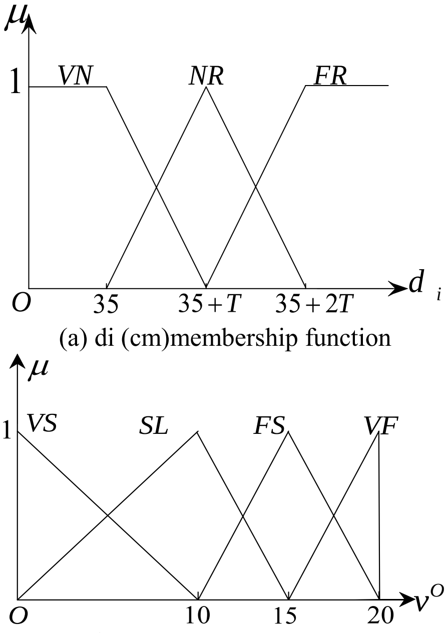
$$\mu_{V^{G'}}(v^G) = \bigcup_{j=1}^{28} \mu_j(\phi', d_g') \wedge \mu_{V_j^G}(v^G) \tag{8}$$

$$\mu_{\omega^{G'}}(\omega^G) = \bigcup_{j=1}^{28} \mu_j(\phi', d_g') \wedge \mu_{\omega_j^G}(\omega^G) \tag{9}$$

B.obstacle avoider

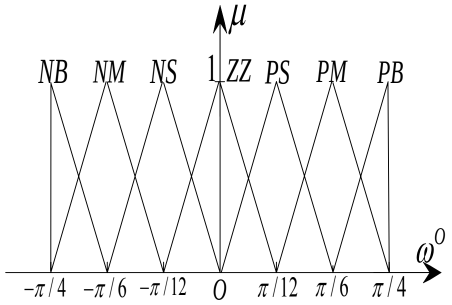
The simple obstacle-avoiding behavior only consider whether the robot can avoid the obstacles in the environment in real time, regardless of deviating from the goal direction. This obstacle-avoiding behavior can help mobile robot to move freely without collision in the unknown environment. This Obstacle Avoider is also implemented through a fuzzy controller. The input variable is d_i (i = 1,2,3,4,5), and is fuzzified into a fuzzy set with three fuzzy languages {VN, NR, FR}, i.e. VN: very near (35), NR: near (35+T), FR: far (35+2T). The domain of the fuzzy set is determined by the variable T, and we can let the robot adapt to different specific circumstances by adjusting the variable T. Its membership functions are shown in Figure 7 (a). The output variables of the controller are $v^{O,}$ and $\omega^{O.}$ v^O is fuzzified into a fuzzy set with four fuzzy languages {VS, SL, FS, VF} (VS: very slow ,

SL: slow, FS: fast, VF: Very fast) as shown in Figure 7 (b). Output variable ω^O is fuzzified into a fuzzy set with seven fuzzy languages {NB, NM, NS, ZZ, PS, PM, PB} as shown in Figure 7 (c).



(a) di (cm)membership function

(b) v^O (cm/s) membership function



(c) ω^O (rad) membership function

Figure 7. Fuzzification of obstacle-avoiding behavior

According to the actual obstacle-avoiding experience of a motor vehicle, a rule base can be roughly summed up as a robot moves toward direction without obstacle: under the premise that there are obstacles in the front, if there are not obstacles in the left front, then turn slightly to the left, and vice versa if there is no obstacle in the left, then turn left with magnitude of rotation; if there are not obstacles in right front, then turn slightly to the right, and vice versa if there are no obstacles in the right, then turn to the right with big magnitude of rotation; the closer to the obstacle, the smaller the line speed will be, and the further to the obstacle, the bigger the line speed will be.

Since there are five fuzzy inputs d_1,d_2,d_3,d_4,d_5, and each membership function has three fuzzy languages, therefore, there is a total of 3^5=243 fuzzy rules in the fuzzy rule base, for example:

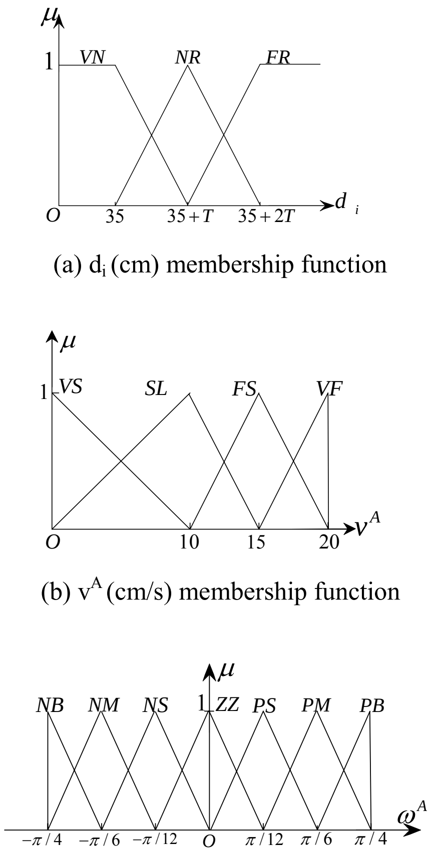
Rules1: if d_1 is VN and d_2 is VN and d_3 is VN and d_4 is VN and d_5 is VN then v^O is VS and ω^O is NB;

Rules2: if d_1 is FR and d_2 is FR and d_3 is FR and d_4 is FR and d_5 is FR then v^O is VF and ω^O Is ZZ;

C. Obstacle following behavior controller

The purpose to design obstacle following behavior is to enable the robot to achieve real-time tracking of obstacles based on the real need of navigation, especially when the robot is trapped in local minimum (for example, when the robot falls into a U-shaped groove obstacle) of path planning. Obstacle following behavior can make the robot run along the edge of the wall closer to it, and help the robot to escape from local minima traps so as to solve the local minimum in path planning. In general, obstacle following behavior can be subdivided into walking along the left edge and walking along the right edge. The input variable of this fuzzy controller is di (i=1,2,3,4,5), which is the same as the input variable of the Obstacle Avoider; output variables are vA and ωA. The membership functions of input and output variables are shown in Fig. 8. There are 5 fuzzy inputs d1, d2, d3, d4, d5, and each membership function has three fuzzy languages.

Thus, there is a total of 3^5=243 fuzzy rules in the rule base. In these rules, some are fuzzy control rules walking along the right of the obstacle. The design principle is that



(a) d_i (cm) membership function

(b) v^A (cm/s) membership function

(c) ω^A (rad) membership function

Figure 8. Fuzzification of behavior to walk along the edge of the obstacle

When the sensor on the right detects the robot is close to the obstacle on the right, it will move toward the left side which is far away from obstacles; when the sensor on the right detects robot is far from the obstacle on the right, it will move toward the obstacle on the right side; and so forth, the robot completed the movement along the right edge of the obstacle in the dynamic balance of moving away from the obstacle and moving toward the obstacle, such as fuzzy rules can be seen as follows:

Rules1: if d1 is VN and d2 is VN and d3 is NR then v^A is VS and ω^A is PB.

Rules2: if d1 is NR and d2 is NR and d3 is FR then v^A is SL and ω^A is PM.

In these rules, some are fuzzy control rules walking along the left edge of the obstacle, such as:

Rules12: if d5 is VN and d4 is VN and d3 is NR then v^A is VS and ω^A is NB.

Rules13: if d5 is NR and d4 is NR and d3 is FR then v^A is SL and ω^A is NM.

D. Behavior weighting controller

The function of Behavior Weighting Controller is to get the weighting factor with integration of three behaviors OA, GS, OFBC, so as to obtain collision-free path planning routes. Here, set p as the number of corresponding sensor along the goal direction,

$$k = 8 - \text{int}(\frac{\phi}{15}) \qquad (10)$$

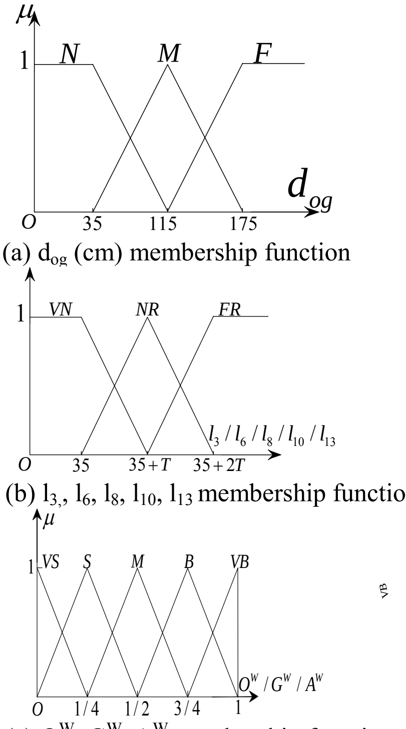
Then

$$p = \begin{cases} k \quad , k \geq 1 \\ k + 24, k < 1 \end{cases} \qquad (11)$$

Set d_{og} as obstacle distance in the goal position vector, then:

$$d_{og} = R_V + \min(l_i \mid i = p-1, p, p+1) \; . \qquad (12)$$

The input variables of the fuzzy controller are d_{og} and l_3, l_6, l_8, l_{10}, l_{13}. The output variables are O^W, G^W, A^W. Variable d_{og} is fuzzified into a fuzzy set with three fuzzy languages {N, M, F}. Variables l_3, l_6, l_8, l_{10}, l_{13} are fuzzified into a fuzzy set with three fuzzy languages {VN, NR, FR}. Output variables O^W, G^W, A^W are fuzzified into a fuzzy set with five fuzzy languages {VS, S, M, B, VB}.The membership functions of fuzzy input and output variables are as shown in Fig. 9.



(a) d_{og} (cm) membership function

(b) l_3,, l_6, l_8, l_{10}, l_{13} membership function

(c) O^W, G^W, A^W membership function

Figure 9. Fuzzification of variables in Behavior Weighting Controller

Because there are six fuzzy input d_{og}, l_3, l_6, l_8, l_{10}, l_{13}, and each membership function has three vague languages, therefore, there are a total of 3^6=729 fuzzy rules in the fuzzy rule base. For example:

Rules1: if d_{og} is F and l_3 is FR and l_6 is FR and L_8 is FR and l_{10} is FR and l_{13} is FR then O^W is VS and G^W is VB and A^W is VS.

Rules17: if d_{og} is M and l_3 is NR and l_6 is FR and l_8 is VN and l_{10} is FR and l_{13} is NR then O^W Is VB and G^W is VS and A^W is VS.

Rules249:if d_{og} is N and l_3 is VN and l_6 is VN and l_8 is VN and l_{10} is NR and l_{13} is FR then O^W is VS and G^W is VS and A^W is VB.

E. Multi-behavioral fusion

The output drawn by Obstacle Avoider (OA), Goal Seeker (GS) and Obstacle Following Behavior Controller (OFBC) for the path planner will be respectively conducted with weighted handling through output weighting of Behavior Weighting Controller (BW), then get the total output V and ω after commanding fusion, as shown in Fig. 5.

Finally, the gravity method is used for defuzzification and obtain precise amount of control V and ω:

$$V = \frac{G^W \sum_{j=1}^{28} \mu_j(\phi', d_g')k_j + O^W \sum_{m=1}^{243} \mu_m(d')r_{1m} + A^W \sum_{n=1}^{243} \mu_n(d'')z_{1n}}{G^W \sum_{j=1}^{28} \mu_j(\phi', d_g') + O^W \sum_{m=1}^{243} \mu_m(d') + A^W \sum_{n=1}^{243} \mu_n(d'')} \tag{13}$$

$$\omega = \frac{G^W \sum_{j=1}^{28} \mu_j(\phi', d_g')q_j + O^W \sum_{m=1}^{243} \mu_m(d')r_{2m} + A^W \sum_{n=1}^{243} \mu_n(d'')z_{2n}}{G^W \sum_{j=1}^{28} \mu_j(\phi', d_g') + O^W \sum_{m=1}^{243} \mu_m(d') + A^W \sum_{n=1}^{243} \mu_n(d'')} \tag{14}$$

Wherein, in formula (13):

$$\mu_j(\phi', d_g') = \mu_{\Phi_j}(\phi') \wedge \mu_{D_j}(d_g') \tag{15}$$

$$\begin{aligned} &\mu_m(d') = \mu_m(d_1', d_2', d_3', d_4', d_5') \\ &= \mu_{D_m}(d_1') \wedge \mu_{D_m}(d_2') \wedge \mu_{D_m}(d_3') \wedge \mu_{D_m}(d_4') \wedge \mu_{D_m}(d_5') \end{aligned} \tag{16}$$

$$\begin{aligned} &\mu_n(d'') = \mu_n(d_1'', d_2'', d_3'', d_4'', d_5'') \\ &= \mu_{D_n}(d_1'') \wedge \mu_{D_n}(d_2'') \wedge \mu_{D_n}(d_3'') \wedge \mu_{D_n}(d_4'') \wedge \mu_{D_n}(d_5'') \end{aligned} \tag{17}$$

K_j represents the center of No. j fuzzy set in v^G, r_{1m} represents the center No. m fuzzy set in v^O, z_{1n} represents the center of No. n fuzzy set in v^A.

IV. Algorithm Simulation and Experimental Result Analysis

We verify the path planning algorithm proposed in this paper from two aspects: on the one hand, develop a mobile robot path planning algorithm simulation platform in the use of Matlab R2011b for simulation verification; on the other hand, move the robot system in the use of real object for physical verification.

In the mobile robot path planning algorithm simulation platform, the Fuzzy logic toolbox is used for simulation of fuzzy algorithm; in path planning simulation, the start and end position can be arbitrarily set, and any amount of rectangular barriers can be placed anywhere.

As provided in Fig. 10, eight obstacles are set in the movement environment of the robot. The physical robot moves from the top to the bottom, well ending the route planning process. It can be seen the path planning is very smooth and fluid.

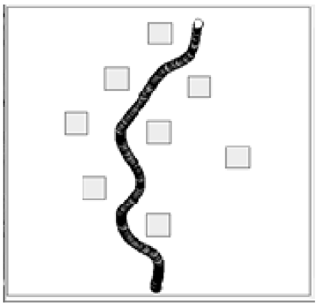
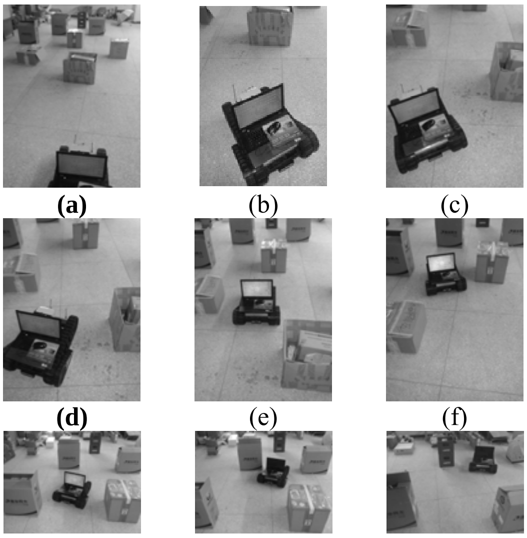


Figure 10. Mobile robot path planning algorithm simulations



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 11. The navigation process of physical robot

The physical environment as shown in Fig. 11 matches the environment shown in Fig. 10. The robot moved from the start point to the corresponding destination point, and successfully completed the entire navigation process.

The above simulation and experimental results show that the physical robot can safely avoid obstacles and reach the goal point in the use of this algorithm with high stability and robustness.

In the U-shaped groove obstacle defined in Fig. 12, the robot could walk along the edge of the obstacle to flee U-groove trap, and successfully reach the goal point from the start point.

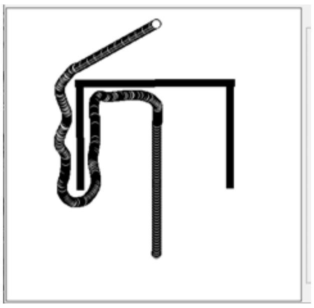
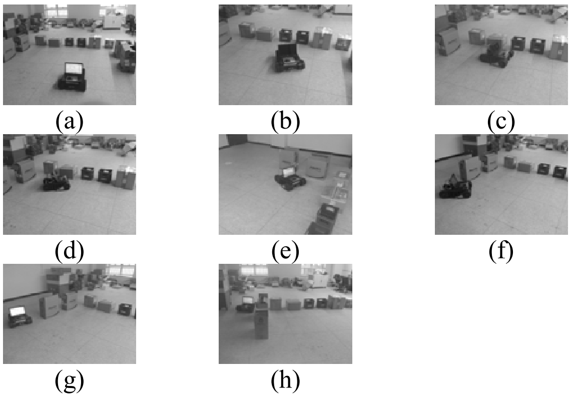


Figure 12. Simulation of escaping from U-groove

Fig. 13 is the navigation process of the corresponding physical robot in the above path planning.



(a) (b) (c)

(d) (e) (f)

(g) (h)

Figure 13. Navigation process of the robot escaping from U-groove

According to the above simulation and experiment, we can draw that this path planning algorithm can solve the local minimum problem in a similar U-shaped groove obstacle.

V. Conclusion

This paper presents a fuzzy logic path planning algorithm based on multi behavior. First use RFID technology to achieve precise positioning of the mobile robot, and use fuzzy logic control technology to create the Goal Seeker (GS), Obstacle Avoider (OA) and Obstacle Following Behavior Controller (OFBC) etc. three basic behaviors. The new multi-behavioral fusion algorithm with stability, high efficiency, and robustness can be achieved in the Behavior Weighting Controller (BW) to solve the local minimum problem in obstacle such as a U-shaped groove in the environment that cannot be solved with the fusion of goal-trending and obstacle-avoiding behaviors, also to solve the problems in traditional algorithms such as highly dependent on the environmental information, necessary to establish an environment map, poor real time, low robust in the algorithm. The behavior of mobile robot shows good consistency, high robustness, continuity and stability.

Acknowledgment

This research was financially supported by the key special project of Chongqing robotics industry technology (cstc2013jcsf-zdzxqqX0005) and Funding project of Chongqing Quality and Technical Supervision Bureau (CQZJKY2014024).

References

[1] Bai Jinke, Research on mobile robot path planning, Henan: Physics and Electronics College of Henan University, 2011.

[2] Luo Qianyou, Zhang Hua, Wang Heng etc., Improve the application of artificial potential field in robot path planning, Computer Engineering and Design, vol.32 , pp. 1411-1413, 2011.

[3] Liu Yu, Lu Jianfeng, Cai Haizhou, Research on robot path planning based on A* algorithm, Computer Technology and Development, vol.22, pp. 108-111, 2012.

[4] Zhu Lei, Fan Jizhuang, Zhao Jie etc., Robot searching mine accident global path planning and local obstacle avoidance based on grid method, Journal of Central South University (Natural Science), vol.42 , pp. 3421-3428, 2011.

[5] Gao Jingxin, Ruan Xiaogang, Ma Shengce, Proactive routing navigation of robot based on fuzzy behavior decision, Journal of Beijing University of Technology, vol.40, pp. 1308-1314, 2014.

[6] Cai Chengtao, Zhu Qidan, Mobile robot path planning simulation based on fuzzy controller, Computer Simulation, vol.25, pp. 182-186, 2008.

[7] Qing Li, Path planning based on fuzzy logic algorithm for mobile robots in static environment, Control and Decision Conference (CCDC), 2013 25th Chinese: pp.2866-2871, 2013.

[8] Zhu Daqi, Yan Mingzhong, Technical review of mobile robot path planning, Control and Decision,vol. 25, pp.961-967, 2010.

[9] Chen Weifeng, Xue Dong, Zhou Wangping, Mobile robot path planning based on fuzzy control of behavior, Computer Measurement & Control, vol.22 , pp. 3600-3602, 2014.

[10] Ma Jiachen, Zhang Qi, Ma Liyong etc., Multi-behavior fusion-based path planning for mobile robot, Journal of Beijing Institute of Technology, vol.34, pp.576-581, 2014.

[11] Zhang Xiuhua, Mobile robot autonomous positioning algorithm based on WLAN/RFID Information fusion, Harbin: College of Mechanical Power Engineering, Harbin University of Science and Technology, 2013.