

Navigation System of Mobile Robot in an Uncertain Environment Using Type-2 Fuzzy Modelling

Sittichok Junratanasiri

Computer Engineering Department,
Faculty of Engineering, Chiang Mai
University,
Chiang Mai, Thailand
dsj_4@hotmail.com

Sansanee Auephanwiriyaikul

Computer Engineering Department,
Faculty of Engineering, Chiang Mai
University,
Chiang Mai, Thailand
sansanee@ieee.org

Nipon Theera-Umpon

Electrical Engineering Department,
Faculty of Engineering, Chiang Mai
University,
Chiang Mai, Thailand
nipon@ieee.org

Abstract—A navigation system of a mobile robot in an uncertain environment is one of a popular research area. There are both static and dynamic obstacles in an uncertain environment. If a robot finds a static obstacle, it can select a safe path very easily. However, a dynamic obstacle can move very randomly in the global map. Hence, in this paper, we propose a navigation system in an uncertain environment focusing on dynamic obstacles for a mobile robot. The future position of a dynamic obstacle is modeled using a fuzzy vector. The dangerous region of that obstacle is then computed. Then the free road candidates are computed based on those dangerous regions found. Then the best free road is selected. Finally, the interval type-2 fuzzy logic system is utilized to compute the velocity and angular velocity of a mobile robot. The experiment results show that our navigation system worked in an uncertain environment, i.e., an environment with an obstacle with stable velocity and angular velocity, an environment with an obstacle with random velocity, and an environment with several obstacles with both condition.

Keywords—interval type-2 fuzzy logic; uncertain environment; fuzzy collision region; collision prediction, obstacle avoidance

I. INTRODUCTION

One of the popular research areas in robot navigation system is to navigate in an unknown or partially unknown dynamic environment. There are both static and dynamic obstacles with uncertainty in position, velocity and angular speed in the dynamic environment. There are several cause of uncertainty, e.g., limitation of the perception system, an incorrect or incomplete data from sensor, and etc. Usually, the navigation algorithm needs to have abilities to avoid collision with obstacles, to find free road to the goal, and self-localization. A dynamic obstacle position prediction is needed in obstacle avoidance algorithm. To predict future position of a dynamic obstacle, the estimation of the position and velocity of each object are needed.

There are several research works on modeling the motion of the obstacles and dangerous region with uncertainty using a probabilistic distribution [1 – 3]. Then a collision prediction technique is added to improve local path planning algorithm. However, in [1, 3], only linear velocity obstacles are modeled.

In [4, 5], the potential field method is used for obstacle avoidance in robot navigation. The optimal and minimum occupancy path is generated. In [6], the flexible vector field is used for obstacle avoidance. The path found by [6] is shorter than the one from [4]. However, both navigation methods cannot find a good path if there are dynamic obstacles in the scene. In [7], they use radial-basis-function-neural network to predict the movement of the obstacle. However, the movement of the obstacle has to fall into one of the patterns that the system is trained on.

Type-1 fuzzy logic has been used in robotic navigation and control [2, 4, 8 – 11]. Because the system is able to control nonlinear, ill-defined problem and can make a decision under uncertainty environment. Recently, a type-2 fuzzy logic becomes a popular method in controlling a robot [12 – 14] when the fuzzy membership function is hard to define.

This paper presents an autonomous robot navigation algorithm for a mobile robot working in a dynamic environment. In particular, the velocity, angular velocity and position uncertainties of an obstacle are modeled using fuzzy membership functions. Also, the dangerous region computed from future collision region of each obstacle is modeled using a fuzzy vector. After, all dangerous regions are found, the best path is computed. Then the type-2 fuzzy logic system is used to controlled velocity and angular velocity of a robot.

II. INTERVAL TYPE-2 FUZZY LOGIC SYSTEMS

In interval type-2 fuzzy logic system [15], the crisp inputs are mapped into type-2 fuzzy singleton shown in fig. 1. The upper ($\bar{f}$) and lower ($\underline{f}$) membership function is computed. Suppose there are M rules and there are two inputs and one output for each rule. Each rule is in the form of: "If x_1 is F_1 and x_2 is F_2 then y is G " where F_1 , F_2 and G are interval type-2 fuzzy sets in linguistic terms of antecedents and consequent, respectively. The minimum t-norm is used to combine the upper and lower membership values of antecedents as shown in Fig. 2 whereas the maximum t-conorm is utilized to aggregate outputs from all rules that are fired. In order to get crisp output, the type reduction and defuzzification is needed.

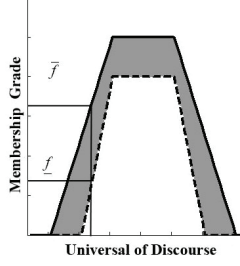


Figure 1. Interval singleton type-2 fuzzy logic system

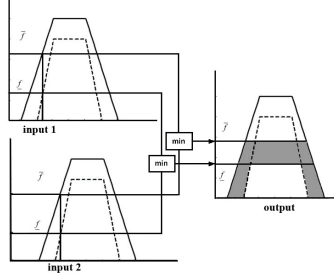


Figure 2. Inputs and antecedent operator for an interval single type-2 fuzzy logic system.

III. MODEL OF UNCERTAINTY ENVIRONMENT

A. Mobile Robot

In this paper we assume that the robot is a circle with center in (x_R, y_R) when the robot axis is (X_R, Y_R) and Y_R is the orientation of the robot. Whereas the counter clockwise is the positive direction and the angle range of this coordinates is $[0, 180]$. We assume that the robot can see within the range of vis .

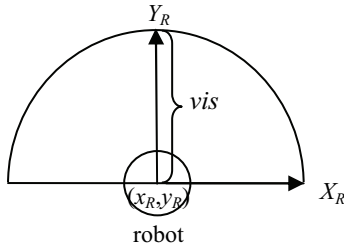


Figure 3. The robot is a circle with centre in (x_R, y_R) and (X_R, Y_R) is Robot axis.

B. Obstacle Modelling

There are both static and dynamic obstacles in an uncertain environment. The mobile robot moves through the environment with its perception setting. In real environment, the position of a static obstacle does not change. We directly use information from the perception system. Hence, the dangerous region in this case is the position of the static obstacle. But dynamic obstacle can change its position with a random move. Therefore, a fuzzy membership function is used to describe the uncertainty in position, velocity, and angular velocity of dynamic obstacles. For simplicity, we assume that the mobile robot can move faster than a dynamic obstacle in this environment. We also assume that the shape of each obstacle is circular. Fig. 4 shows a mobile robot moving in a dynamic

environment and a dynamic obstacle i is detected by this robot. The dangerous region is the predicted collision region in the future.

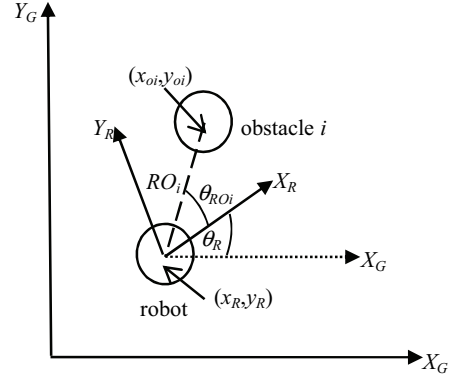


Figure 4. A dynamic obstacle i is detected

To predict a collision region, we need to estimate the position of a dynamic obstacle. Suppose that the current center of the dynamic obstacle i is $(x_{oi}(t_c), y_{oi}(t_c))$. It is computed based on the global map (X_G, Y_G) at time t_c . The center point can be expressed as:

$$x_{oi}(t_c) = x_R(t_c) + RO_i \cos(\theta_R(t_c) + \theta_{ROi}(t_c)) \quad (1)$$

$$y_{oi}(t_c) = y_R(t_c) + RO_i \sin(\theta_R(t_c) + \theta_{ROi}(t_c)) \quad (2)$$

where RO_i denotes the distance between the center of the obstacle i and the robot, θ_{ROi} is the relative angle of its center and the robot and $\theta_R(t_c)$ denotes the orientation angle of the robot relative to the global map at time t_c .

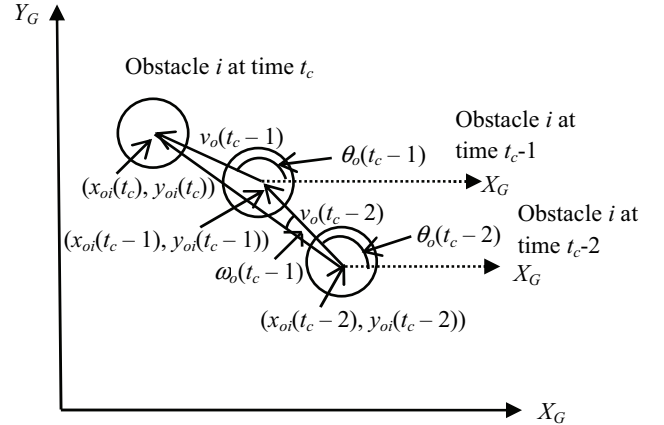


Figure 5. The velocity of obstacle i at time $t_c - 1$ and $t_c - 2$ and angular velocity of obstacle i at time $t_c - 1$

Fig. 5 shows the velocity, angular velocity and angle relative with the global map of a dynamic obstacle i at time $t_c - 1$ and $t_c - 2$. If there is an information of the position of a dynamic obstacle at time $t_c - 1$ and t_c , the velocity $v_o(t_c - 1)$ can be calculated. The angular velocity $\omega_o(t_c - 1)$ can be computed if the position of the dynamic obstacle at time $t_c - 2$, $t_c - 1$ and t_c are known.

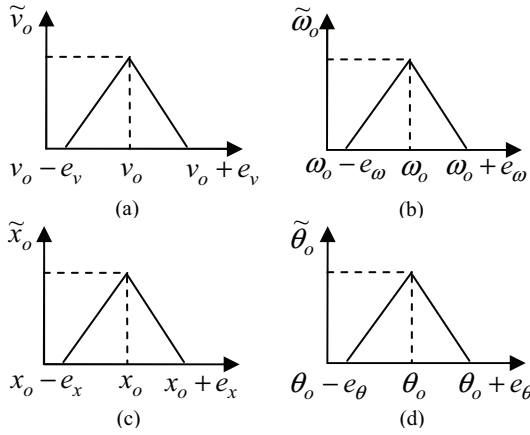


Figure 6. Fuzzy membership function of (a) velocity, (b) angular velocity, (c) position in x or y axis (d) angle relative with global map

To get the better trend of the direction and magnitude of the velocities, the velocity $v_o(t_c)$ and angular velocity $\omega_o(t_c)$ of dynamic obstacle i are obtained from the calculation of three previous times ($t_c - 1, t_c - 2$ and $t_c - 3$), i.e.,

$$v_o(t_c) = \lambda_1 v_o(t_c - 1) + \lambda_2 v_o(t_c - 2) + \lambda_3 v_o(t_c - 3) \quad (3)$$

$$\omega_o(t_c) = \lambda_1 \omega_o(t_c - 1) + \lambda_2 \omega_o(t_c - 2) + \lambda_3 \omega_o(t_c - 3) \quad (4)$$

where λ_1, λ_2 and λ_3 are weight parameters. In the experiment, we set the parameters to 0.5, 0.3 and 0.2 respectively.

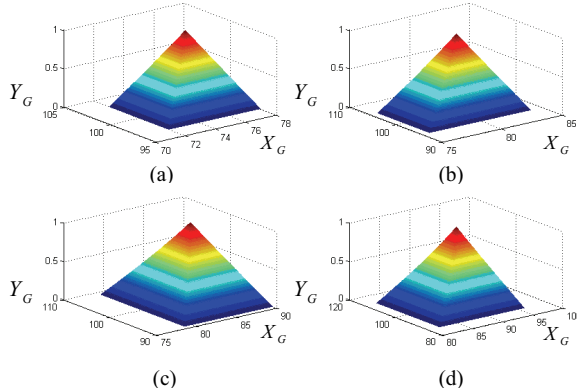


Figure 7. Fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$ at time (a) t_c+1 , (b) t_c+2 , (c) t_c+3 and (d) t_c+4

Since there are uncertainty in velocity (v_o) and angular velocity (ω_o) of each dynamic obstacle, we model the uncertainty with triangular fuzzy membership functions for both velocities shown in figures 6(a) and (b). We also model the position and the angle (θ_o) of each dynamic obstacle relative with the global map with triangular fuzzy membership functions shown in figures 6(c) and (d).

The fuzzy sets $\tilde{v}_{oi}, \tilde{\omega}_{oi}$ denote velocity and angular velocity fuzzy sets, whereas $(\tilde{x}_{oi}, \tilde{y}_{oi})$ denotes a position fuzzy vector and $\tilde{\theta}_{oi}$ denotes orientation fuzzy set of dynamic

obstacle i . We can computed the position fuzzy vector at time t' by

$$\tilde{x}_{oi}(t') = \tilde{x}_{oi}(t_c) + (\tilde{v}_{oi}(t_c) \cos(\tilde{\theta}_{oi}(t_c) + \tilde{\omega}_{oi}(t_c) \Delta t)) \Delta t \quad (5)$$

$$\tilde{y}_{oi}(t') = \tilde{y}_{oi}(t_c) + (\tilde{v}_{oi}(t_c) \sin(\tilde{\theta}_{oi}(t_c) + \tilde{\omega}_{oi}(t_c) \Delta t)) \Delta t \quad (6)$$

$$t' = t_c + \Delta t \quad (7)$$

To make the calculation of equations (5) and (6) simple, the lower and upper bound of the support and the core of each fuzzy set is used to compute the lower and upper bound of the support and the core of $\tilde{x}_{oi}$ and $\tilde{y}_{oi}$ in equations (5) and (6).

Fig. 7 shows the fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$ at time $t_c + 1$ to $t_c + 4$. Then the dangerous region is computed by extending the support of the fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$ at time t' with the radius of obstacle i as shown in fig. 8.

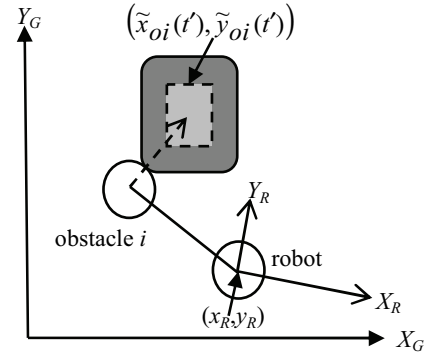


Figure 8. The gray area inside the solid line is the dangerous region whereas the one inside the dashed line is computed from the support of a fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$.

IV. COLLISION PREDICTION TECHNIQUE

The collision prediction technique is used to predict the dangerous region (a collision region for dynamic obstacles). Since the dangerous region is a fuzzy vector, the predicted collision time is assumed to be a triangular fuzzy set $\tilde{t}_{col}$ with parameter $(Tcola, Tcolb, Tcolc)$ as shown in fig. 9. The lower bound of the support of $\tilde{t}_{col}$ ($Tcola$) is assumed to be the first predicted collision time, whereas the upper bound of the support of that ($Tcolc$) is the last predicted collision time.

The distance between the robot and the obstacle at time $\tilde{t}_{col}$ is

$$\widetilde{disRO_i}(\tilde{t}_{col}) = \sqrt{(x_R(\tilde{t}_{col}) - \tilde{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \tilde{y}_{oi}(\tilde{t}_{col}))^2} \quad (8)$$

For simplicity, we use the lower bound of the support $(\tilde{x}_{oi}, \tilde{y}_{oi})$, the core $(\hat{x}_{oi}, \hat{y}_{oi})$ and the upper bound of the support $(\bar{x}_{oi}, \bar{y}_{oi})$ of fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$ to compute the lower bound $(\widetilde{disRO_i}(\tilde{t}_{col}))$ and upper bound $(\overline{disRO_i}(\tilde{t}_{col}))$

of the support and the core ($\widetilde{disRO_i}(\tilde{t}_{col})$) of $\widetilde{disRO_i}(\tilde{t}_{col})$. Hence, the equation (8) will be

$$\widetilde{disRO_i}(\tilde{t}_{col}) = \sqrt{(x_R(\tilde{t}_{col}) - \tilde{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \tilde{y}_{oi}(\tilde{t}_{col}))^2} \quad (9)$$

$$\widehat{disRO_i}(\tilde{t}_{col}) = \sqrt{(x_R(\tilde{t}_{col}) - \hat{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \hat{y}_{oi}(\tilde{t}_{col}))^2} \quad (10)$$

$$\underline{disRO_i}(\tilde{t}_{col}) = \sqrt{(x_R(\tilde{t}_{col}) - \underline{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \underline{y}_{oi}(\tilde{t}_{col}))^2} \quad (11)$$

Suppose that δ is the shortest distance between the robot and the obstacle. Then, to find $\tilde{t}_{col}$, we can set

$$\widetilde{disRO_i}(\tilde{t}_{col}) = \delta \quad (12)$$

For simplicity, to find $Tcola$, $Tcolb$ and $Tcolc$ of $\tilde{t}_{col}$, we need to solve equations 13, 14 and 15 respectively with the Leap-frogging Newton's method [16].

$$(x_R(\tilde{t}_{col}) - \tilde{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \tilde{y}_{oi}(\tilde{t}_{col}))^2 - \delta^2 = 0 \quad (13)$$

$$(x_R(\tilde{t}_{col}) - \hat{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \hat{y}_{oi}(\tilde{t}_{col}))^2 - \delta^2 = 0 \quad (14)$$

$$(x_R(\tilde{t}_{col}) - \underline{x}_{oi}(\tilde{t}_{col}))^2 + (y_R(\tilde{t}_{col}) - \underline{y}_{oi}(\tilde{t}_{col}))^2 - \delta^2 = 0 \quad (15)$$

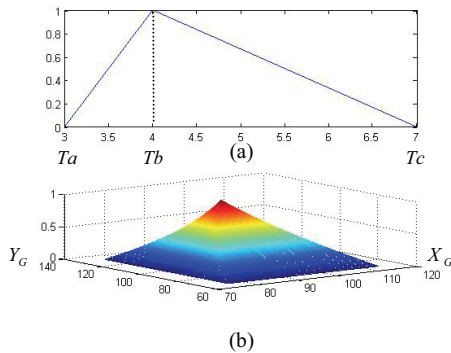


Figure 9. (a) $\Delta \tilde{t}$ of obstacle i (b) the predicted fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$ at time $\tilde{t}_{col}$ of obstacle i

Then, we can compute Δt in equation (7). Since $\tilde{t}_{col}$ is a triangular fuzzy set, $\Delta \tilde{t}$ will be a triangular fuzzy set with parameter (Ta, Tb, Tc) as shown in fig. 9(a). The fuzzy vector $(\tilde{x}_{oi}, \tilde{y}_{oi})$ at time $\tilde{t}_{col}$ (shown in fig. 9(b)) can be computed from equations (5) and (6). Again, for simplicity, we use the lower bound, the upper bound of the support and the core of the input fuzzy sets to compute the lower bound, the upper bound of the support and the core of $\tilde{x}_{oi}$ and $\tilde{y}_{oi}$ in equations (5) and (6).

V. NAVIGATION PLANNING

When all dangerous regions are calculated by collision prediction technique, the robot needs to find the free road that does not intersect with any dangerous regions. In this paper, the finding process is divided into finding the free road candidate (ϕ_{RC}) according to the axis X_R from free sector (all safe regions in global map) and choosing the best free road (ϕ_{RS}) using a costing function.

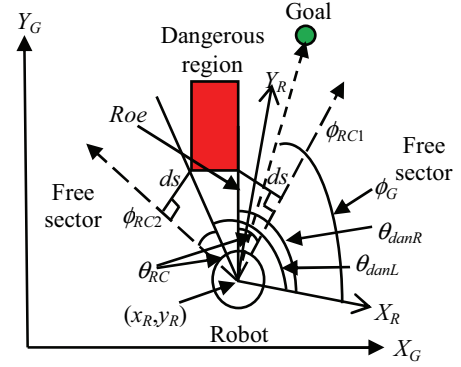


Figure 10. Finding free road candidate when there is 1 obstacle in the path to the goal

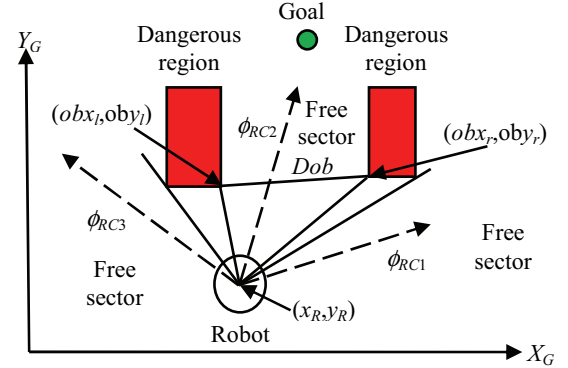


Figure 11. Finding free road candidate when robot moves through a narrow free sector

1) Finding the free road candidate

The location of the goal is the important factor in the finding the free road candidate. Hence, this process is as follows

a) If the goal position (ϕ_G) according to the axis X_R is located in the free sector (not in the dangerous region) and there is no collision with any obstacle in between, then $\phi_{RC} = \phi_G$

b) If the goal position (ϕ_G) is not located in the free sector or is inside the free sector and there is a collision with any obstacle in between (shown in fig. 10), then

$$\theta_{RC} = \sin^{-1}\left(\frac{ds}{RO_e}\right) \quad (16)$$

where ds is the shortest distance between the dangerous region and the choosing path and RO_e is the shortest distance between the dangerous region and the center of the robot. Suppose that

θ_{danR} and θ_{danL} and are the angles from the axis X_R to the right and left boundaries of the dangerous region, respectively. Then we can compute free road candidate on the right (ϕ_{RC1}) and the left (ϕ_{RC2}) free road by

$$\phi_{RC1} = \theta_{danR} - \theta_{RC} \quad (17)$$

$$\phi_{RC2} = \theta_{danL} + \theta_{RC} \quad (18)$$

c) If the path to the goal is in the free sector that is narrow (shown in fig. 11), then we need to compute the movable distance (Dob), i.e.,

$$Dob = \sqrt{(obx_l - obx_r)^2 + (oby_l - oby_r)^2} \quad (19)$$

where (obx_l, oby_l) and (obx_r, oby_r) are the left and right boundaries of the left and right dangerous region, respectively. Let the safe width (Rs) is bigger than the diameter of the robot, and suppose that $Dob > Rs$, then this path is safe. Hence, we can find ϕ_{RC} according to case a or b.

2) Danger estimation

After the free road candidate ϕ_{RC} is computed and the robot moves following the path ϕ_{RC} , we need to estimate if there is a collision or not. Let $(x_{ro_{near}}, y_{ro_{near}})$ be the center of the nearest dangerous region located in the chosen sector. We can compute $(x_{ro_{near}}, y_{ro_{near}})$ using equations (5) and (6). Let v_R is the velocity of the robot. Then we can compute the position of the robot (x_R, y_R) at time T_{nc} (collision time) using

$$x_R(T_{nc}) = x_R + (v_R \cos(\theta_R + \phi_{RC})) \Delta T_{nc} \quad (20)$$

$$y_R(T_{nc}) = y_R + (v_R \sin(\theta_R + \phi_{RC})) \Delta T_{nc} \quad (21)$$

where ΔT_{nc} is the different time between the current and the collision time. However, T_{nc} is computed using the same method as in section IV. But since we only need this information to estimate the danger in the future of the chosen path, we will use only the core of the nearest dangerous region. Then we can estimate danger (Dv) of ϕ_{RC} by

$$Dv(\phi_{RC}) = \begin{cases} 1 & \Delta T_{nc} < T_{safe} \\ \frac{T_{ignore} - \Delta T_{nc}}{T_{ignore}} & T_{safe} \leq \Delta T_{nc} < T_{ignore} \\ 0 & \Delta T_{nc} \geq T_{ignore} \end{cases} \quad (22)$$

where T_{safe} is the grace period of the stopping time of the robot, T_{ignore} is the threshold of the system.

3) Selection the best free road

The best free road must be the one with the shortest and the safest one. Hence the cost function is

$$COST(\phi_{RC}) = k_1 \cdot \Delta(\phi_{RC}, \phi_G) + k_2 (Dv(\phi_{RC})) \quad (23)$$

where $\Delta(\phi_{RC}, \phi_G)$ is the difference between ϕ_{RC} and ϕ_G , k_1 and k_2 are the weight parameters. In the experiment, we set k_1 and

k_2 to 0.5 and 1, respectively. Then the best free road ϕ_{RS} is the one that give the lowest $COST(\phi_{RC})$.

VI. DESIGN OF INTERVAL TYPE-2 FUZZY LOGIC

There are several researches on utilizing interval type-2 fuzzy logic as a tool to control mobile robot including the works done by [17 – 19]. Most of those research results show that the interval type-2 fuzzy logic controller provide better results than type-1 fuzzy logic controller. Hence, in our experiment, we choose to use the interval type-2 fuzzy logic to control our mobile robot.

There are 3 inputs, i.e., D_{angle} , D_v and v_R . D_{angle} is the difference between ϕ_{RS} and Y_R , i.e., $D_{angle} = \Delta(\phi_{RS}, 90)$ as shown in fig.12. There are 2 outputs, i.e., velocity controller parameter (CV_R) and angular velocity controller parameter (CA_R). All of the membership functions utilized in the experiments are manually picked.

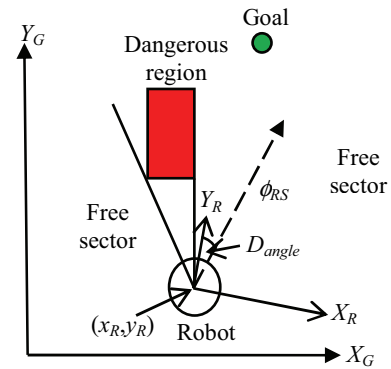


Figure 12. Example of D_{angle}

Figures 13, 14 and 15 show the interval type-2 membership functions of D_{angle} , D_v and v_R . The interval type-2 membership functions of (CV_R) and (CA_R) are shown in figures 16 and 17. The rules used for computing CV_R and CA_R are shown in tables I and II, respectively.

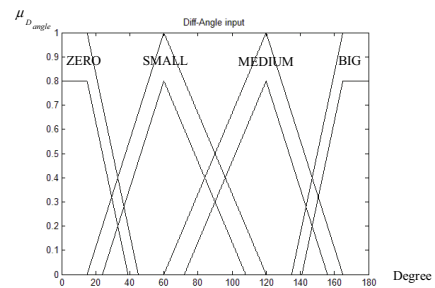


Figure 13. D_{angle} membership functions

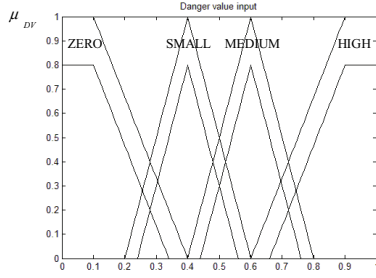


Figure 14. D_v membership functions

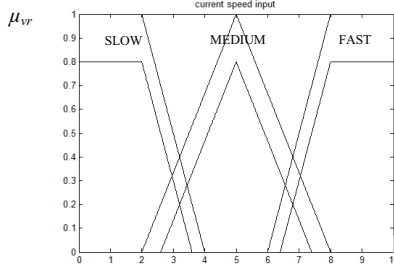


Figure 15. v_R membership functions

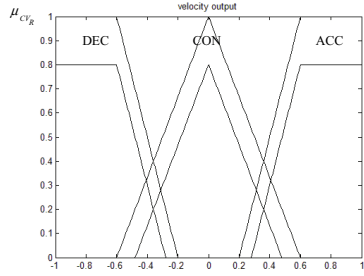


Figure 16. CV_R membership functions

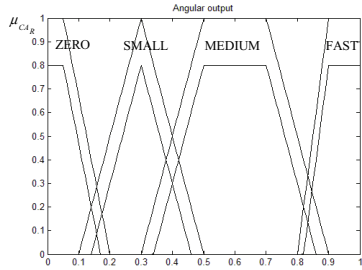


Figure 17. CA_R membership functions

Let ccv_R and cca_R are the defuzzified values of CV_R and CA_R . The velocity and angular velocity of the time $t_c + 1$ is computed by

$$v_R(t_c + 1) = v_R(t_c) + ccv_R \cdot \alpha \quad (24)$$

$$\omega_R(t_c + 1) = \begin{cases} cca_R \cdot \omega_{RMAX} & d_{angle} > 0 \\ -cca_R \cdot \omega_{RMAX} & d_{angle} < 0 \end{cases} \quad (25)$$

where α is the weight parameter and ω_{RMAX} is the maximum angular velocity. Both v_R and ω_R cannot exceed v_{RMAX} and ω_{RMAX} .

TABLE I. FUZZY RULE BASE

rule	rule base of control velocity			
	Input variable			Output variable
	D_{angle}	D_v	v_R	CV_R
1	ZERO - BIG	MEDIUM,HIGH	FAST	DEC
2	ZERO - MEDIUM	HIGH	MEDIUM	DEC
3	MEDIUM	MEDIUM	MEDIUM	DEC
4	BIG	SMALL - HIGH	MEDIUM	DEC
5	BIG	HIGH	SLOW,FAST	DEC
6	ZERO	ZERO	MEDIUM	CON
7	ZERO	MEDIUM	MEDIUM	CON
8	ZERO	HIGH	SLOW	CON
9	SMALL	SMALL,MEDIUM	MEDIUM	CON
10	SMALL	MEDIUM,HIGH	SLOW	CON
11	SMALL,MEDIUM	ZERO,SMALL	FAST	CON
12	MEDIUM	ZERO,SMALL	MEDIUM	CON
13	MEDIUM	SMALL - HIGH	SLOW	CON
14	BIG	ZERO	MEDIUM,FAST	CON
15	BIG	ZERO - MEDIUM	SLOW	CON
16	ZERO	ZERO	FAST	ACC
17	ZERO	ZERO,SMALL	SLOW,MEDIUM	ACC
18	ZERO	MEDIUM	SLOW	ACC
19	SMALL	ZERO,SMALL	SLOW	ACC
20	SMALL	ZERO	MEDIUM	ACC
21	MEDIUM	ZERO	SLOW	ACC

TABLE II. FUZZY RULE BASE

rule	rule base of control angular velocity			
	Input variable			Output variable
	D_{angle}	D_v	v_R	CA_R
1	ZERO	ZERO - HIGH	SLOW - FAST	ZERO
2	SMALL	ZERO	SLOW	ZERO
3	SMALL	ZERO - HIGH	MEDIUM,FAST	SMALL
4	SMALL	SMALL - HIGH	SLOW	SMALL
5	MEDIUM	ZERO - HIGH	SLOW,FAST	MEDIUM
6	MEDIUM	ZERO - MEDIUM	MEDIUM	MEDIUM
7	MEDIUM	HIGH	MEDIUM - FAST	FAST
8	BIG	ZERO - HIGH	SLOW - FAST	FAST

VII. SIMULATION RESULTS

In the experiment, we set v_{RMAX} to 10, ω_{RMAX} is set to $[-1, 1]$, and $ccv_R \cdot \alpha$ is $[-5, 5]$. The diameters of the robot and each dynamic obstacle are 10, whereas vis is set to 75.

A. Unknown dynamic obstacles with stable velocity and angular velocity

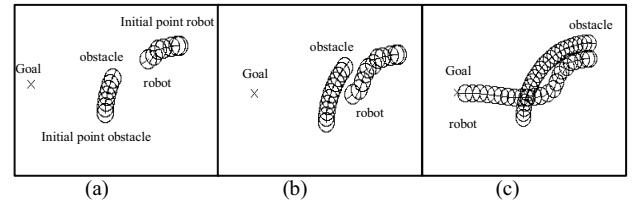


Figure 18. The robot moves to the goal and obstacle moves from bottom to top of the global map (a) both leave from initial, (b) robot decides to avoid obstacle, and (c) robot reaches the goal.

In this scenario, we do not have any information of obstacle. In fig. 18, we set the velocity and angular velocity of the obstacle to 6.3 and -0.07 radius per time step. Fig. 18(a) shows the robot and obstacle in their starting point and their path. Fig. 18(b) shows the robot making a decision to avoid the obstacle based on the proposed method. Then the robot moves very fast to the goal once the obstacle is out of sight of the robot (shown in fig. 18(c)).

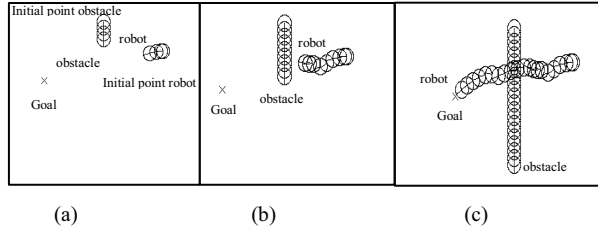


Figure 19. The robot moves to the goal and obstacle moves from top to bottom of the global map (a) both leave from initial, (b) robot decides to avoid obstacle, and (c) robot reaches the goal.

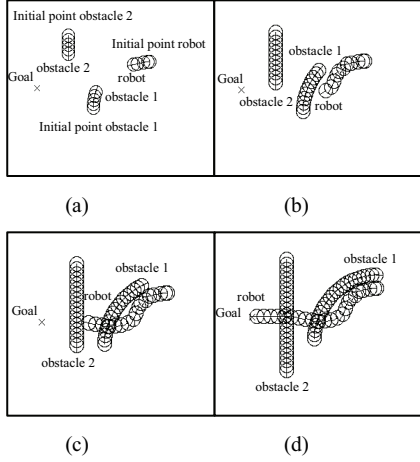


Figure 20. The robot moves to the goal and two obstacles are in the global map (a) robot and obstacles are in the starting point, (b) robot decides to avoid the first obstacle, (c) robot decides to avoid the second obstacle and (d) robot reaches the goal.

Fig. 19 shows the experiment result when the velocity and angular velocity of the obstacle is 7 and 0. Fig. 20 shows the experiment result when there are 2 obstacles with the same setting as in figures 18 and 19.

B. Unknown Dynamic obstacle with unstable velocity and angular velocity

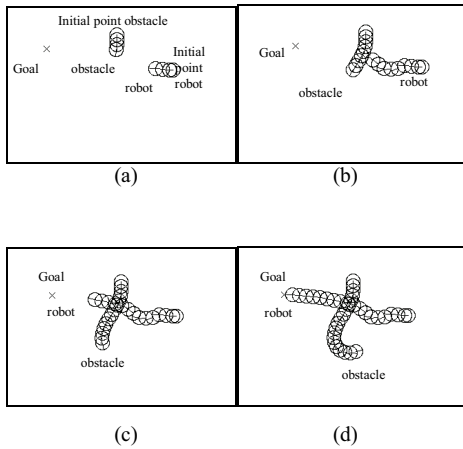


Figure 21. The robot moves to the goal and obstacle moves from top to bottom of the global map (a) robot and obstacles are in the starting point, (b) robot decides to wait for the obstacle to move, (c) the obstacle passes the robot and (d) the robot reaches to goal

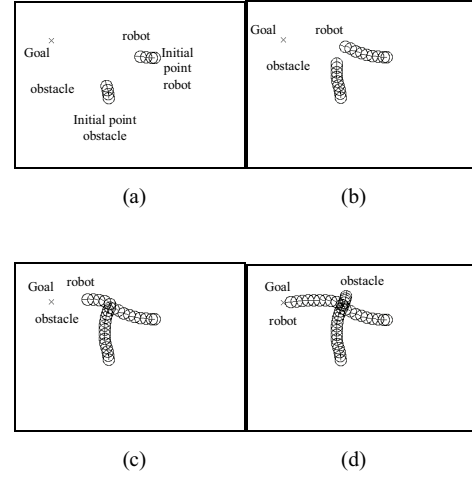


Figure 22. The robot moves to the goal and obstacle moves from bottom to top of the global map (a) robot and obstacles are in the starting point, (b) robot decides to move pass the obstacle, (c) the robot passes the obstacle and (d) the robot reaches to goal

In this experiment, the obstacle moves with random velocity and angular velocity. However, we assume that the robot will move faster than the obstacle. In Fig. 21, the robot decides to wait for the obstacle and then moving towards the goal. Whereas, in Fig. 22, the robot can move pass the obstacle without colliding with the obstacle.

VIII. CONCLUSION

In this work, we propose a navigation system in an uncertain environment (focusing on dynamic obstacle) for a mobile robot. We model the dangerous region for each obstacle by computing its future position fuzzy vector. Then the free road candidates are computed based on those dangerous regions. The best free road is selected based on the shortest and the safest path. Finally the interval type-2 fuzzy logic system is utilized to compute the velocity controller and angular velocity controller of the mobile robot. These parameters are utilized to compute the velocity and angular velocity of the mobile robot. The experiment results show that the mobile robot can find the shortest and safest path in the global map in an uncertain environment, i.e., an environment with an obstacle with stable velocity and angular velocity, an environment with an obstacle with random velocity, and an environment with several obstacles with both condition.

REFERENCES

- [1] Jun Miura and Yoshiaki Shirai, "Modeling Motion Uncertainty of Moving Obstacles for Robot Motion Planning" *2000 IEEE Int. Conf. on Robotics and Automation*, pp. 2258-2263.
- [2] Y. Gao and S.D. Sun, "A collision predict based local path planning of Mobile Robots", *2009 International Asia Conference on Informatics in Control, Automation and Robotics*, pp.185-190
- [3] C. Fulgenzi, A. Spalanzani and C. Laugier, "Dynamic Obstacle Avoidance in uncertain environment combining PVOs and Occupancy Grid", *Proceedings of IEEE International Conference on Robotics and Automation*, 2007
- [4] N.C. Tsourveloudis, K.P. Valavanis and T. Hebert, "Autonomous Vehicle Navigation Utilizing Electrostatic Potential Fields and Fuzzy Logic", *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 4, August 2001

- [5] H. Seki, S. Shibayama, Y. Kamiya and M. Hikizu, "Practical Obstacle Avoidance Using Potential Field for a Nonholonomic Mobile Robot with Rectangular Body", *ETFA 2008*, pp. 326 – 332.
- [6] Y.J. Cho, J.P. Hong and K.W. Park, "Obstacle Avoidance using Active Window and Flexible Vector Field with a Laser Range Finder", *International Conference on Control, Automation and Systems 2007*, Oct. 17-20, 2007 in COEX, Seoul, Korea
- [7] Y. Li, C. Li and R. Song, "A new hybrid algorithm of dynamic obstacle avoidance based on dynamic rolling planning and RBFNN", *Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics*, December 15-18, 2007, Sanya, China
- [8] P. Rusu, E.M. Petriu, T.E. Whalen, A. Cornell and H.J.W. Spoelder, "Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation", *IEEE transactions on instrumentation and measurement*, Vol. 52, No. 4, August 2003
- [9] R. Malhotra and A. Sarkar, "Development of a fuzzy logic based mobile robot for dynamic obstacle avoidance and goal acquisition in an unstructured environment", *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, California, USA, 24-28 July, 2005
- [10] F. Abdessermed, K. Benmahammed and E. Monacelli, "A Fuzzy-based reactive controller for a non-holonomic mobile robot", *Robotics and Autonomous Systems* 47, 2004, pp. 31-46.
- [11] X. Jiang, Y. Motai and X. Zhu, "Predictive Fuzzy Control for a Mobile Robot with Nonholonomic Constraints", *Proceedings of the 2005 IEEE International Conference on Advanced Robotics, ICAR '05*, 18-20 July, 2005
- [12] S. Nurmaini and S.Z.M. Hashim, "An Embedded Fuzzy Type-2 Controller Based Sensor Behavior for Mobile Robot", 2008 *International Conference on Intelligent Systems Design and Applications*, 26-28 November, 2008.
- [13] P. Phokharatkal and S. Phaiboon, "Mobile Robot Control Using Type-2 Fuzzy Logic System", *Proceedings of the 2004 IEEE Conference on Robotics, Automation and Mechatronics*, Singapore, 1-3 December, 2004
- [14] J.Figueroa, J.Posada, J. Soriano, M. Melgarejo and S. Rolas, "A type-2 fuzzy logic controller for tracking mobile object in the context of robotic soccer games", *IEEE International Conference on fuzzy System*, 2005
- [15] J.M. Mendel, "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Direction", University of Southern California Los Angeles, CA
- [16] A.B. Kasturiarachi, "Leap-frogging Newton's method", *International Journal of Mathematical Education in Science and Technology*, Vol.33, No.4, 2002, pp. 521-527.
- [17] M. Krstic, I. Kanellakopoulos and P. Kokotovic, "Nonlinear and adaptive control design", Wiley-Interscience, 1995.
- [18] H. Hagra, "Type-2 Fuzzy Logic Controllers: A Way Forward for Fuzzy Systems in Real World Environments", in WCCI 2008 Plenary/Invited Lectures, LNCS 5050, J.M. Zurada et al. Eds., Berlin Heidelberg : Springer-Verlag, 2008, pp 181-200.
- [19] R. Martinez, A. Rodriguez, O. Castillo, P. Melin, and L. T. Aguilar, "Optimization of Type-2 Fuzzy Logic Contrllers for Mobile Robots Using Evolutionary Methods", *2009 IEEE International Conference on Systems, Man, and Cybernetics*, San Antonio, USA, October 2009, pp. 4764 – 4769.