Fuzzy Logic Based Navigation for Multiple Mobile Robots in Indoor Environments

Ran Zhao¹, Dong Hwan Lee², and Hong Kyu Lee²

¹Department of Electrical and Electronics Engineering, Korea University of Technology and Education, Cheonan, Korea

²Department of Electrical Engineering, Korea University of Technology and Education, Cheonan, Korea

ljfis

Abstract

The work presented in this paper deals with a navigation problem for multiple mobile robot system in unknown indoor environments. The environment is completely unknown for all the robots and the surrounding information should be detected by the proximity sensors installed on the robots' bodies. In order to guide all the robots to move along collision-free paths and reach the goal positions, a navigation method based on the combination of a set of primary strategies has been developed. The indoor environments usually contain convex and concave obstacles. In this work, a danger judgment strategy in accordance with the sensors' data is used for avoiding small convex obstacles or moving objects which include both dynamic obstacles and other robots. For big convex obstacles or concave ones, a wall following strategy is designed for dealing with these special situations. In this paper, a state memorizing strategy is also proposed for the "infinite repetition" or "dead cycle" situations. Finally, when there is no collision risk, the robots will be guided towards the targets according to a target positioning strategy. Most of these strategies are achieved by the means of fuzzy logic controllers and uniformly applied for every robot. The simulation experiments verified that the proposed method has a positive effectiveness for the navigation problem.

Keywords: Robot navigation, Multiple robots, Fuzzy logic, Indoor environment, Dynamic obstacle

1. Introduction

Mobile robots have vast application prospects in areas including space exploring, factory automation, mining, eliminating dangerous situation, military and service, which can economize the labor force to be engaged in other aspects. In these applications, the navigation problem of mobile robots is one of the most popular issues. Thus, how to detect the surrounding information and finding a safe path for the robot is the first condition of success. In an environment with obstacles, the navigation of a robot together with its path planning is to find the collision-free path from a starting location to a target location. In this field, there are many complex problems need to be solved, which mainly due to the nature of the real word and the uncertainty in the environment, such as unpredictability and incompleteness of the knowledge of environment, reliability of sensors, etc. Therefore, a fuzzy logic can be used as a useful tool to deal with these problems.

A number of approaches for tackling the problem of robot navigation have been presented,

Received: Nov. 8, 2015
 Revised : Dec. 23, 2015
 Accepted: Dec. 24, 2015

Correspondence to: Hong Kyu Lee
 (hongkyu@koreatech.ac.kr)
 ©The Korean Institute of Intelligent Systems

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

such as neural network [1], genetic algorithm [2], artificial vision method [3], vector field histogram (VFH) [4], artificial potential field (APF) approach [5] and curvature velocity method [6] in static environment. Most of these methods are usually used in global path planning and hardly be used in real-time control. Many traditional methods designed for global (static) path planning have also been extended to local (dynamic) path planning, such as roadmap method [7], rolling window method [8], etc. Other real-time control methods applied to static environment have also been developed. Jin [9] studied the navigation problem for a mobile robot in an indoor environment by using a command fusion method to govern the robot motions. Nguyen et al. [10] proposed an adaptive robust fuzzy control scheme for path tracking of a wheeled mobile robot. The methods above can be performed well in static environment, but part of them usually are somewhat flawed in dynamic environment because of the assumptions for the moving obstacles as static in a certain period of time.

Navigation for multiple mobile robots is also another important issue for researchers. Zhao and Zou [11] developed a new finite-time synchronized approach for multiple mobile robots formation control based on terminal sliding mode control principle and system synchronization theory. Zhong et al. [12] established a new Velocity-Change-Space method by using the changes of the speed and direction of the robot's velocity for dynamic motion planning. But all the robots must be measured the size, positions and velocities of obstacles and other robots online. The application of fuzzy logic for multiple robot system has also been studied by researchers for years. Nascimento et al. [13] has studied a formation control problem for multiple robots by using the Takagi-Sugeno type fuzzy automaton. Parhi et al. [14] also studied the navigation problem for multiple robots by using a hybrid rule-based-neuro-fuzzy method. Similarly, an online inter-communication among robots is required. Therefore, come studying the proposed problem in completely unknown environment more seem to be have the necessity.

In our previous work [15] we studied the navigation problem for only one robot in the static environment. In this paper, we focus on the proposed approach for multiple robot system in completely unknown dynamic environment based on fuzzy inference system. By means of the danger judgment behavior achieved by a set of proximity sensors, the robots can avoid both convex and concave or dynamic obstacles, and then seek for the targets by another behavior step by step. In order to avoid the "infinite repetition" or "dead cycle" situations, a state memorizing strategy which is used to record the movement

states for the robots is also proposed.

The rest of this paper is organized as follows: Section 2 presents the formulation and working assumptions for proposed problem. Section 3 describes the navigation strategies used in this paper. In Section 4, we verify the effectiveness of the proposed algorithm by a serious of simulation experiments. Finally, conclusions and discussions are included in Section 5.

2. Problem Formulation and Working Assumptions

The experimentations in this work are mainly simulated with a classic wheeled mobile robot. This robot is supported by two DC motors which installed on the left and right wheel respectively. Both sides have an encoder on each wheel, so that the robot can detect how far each wheel has moved by itself. Usually a gyro sensor is also installed on the robot body for the feedback information and correction of driving direction in practice. The structure of such a robot with circular shape is as shown in Figure 1. Totally there are 16 proximity sensors (ultrasonic sensors) equably arranged around the robot body with the same interval. The sensors which are numbered from s_0 to s_{15} are used to measure distances from the robot to the surrounding obstacles or the other robots.

The data that defines a path element can be denoted as a series of coordinate values defining points. We use the generalized coordinates $[x, y, \theta]^T$ to describe the configuration of every point. In this work, we assume that the robot moves on a flat ground and the inertial effects are neglected. The robot is driven under the condition with no slippage, thus it should follow the non-holonomic constraint as:

$$\dot{x} \sin \theta - \dot{y} \cos \theta = 0. \quad (1)$$

where θ is the angle between the robot's driving direction and x-axis. Measured d as the distance between P_W and P_R , which respectively are the center point between two wheels and the center of robot, then, the kinematic functions of P_R can be given as:

$$\begin{cases} \dot{x} = \frac{v_L + v_R}{2} \cos \theta - d \cos \theta \\ \dot{y} = \frac{v_L + v_R}{2} \sin \theta - d \sin \theta \\ \dot{\theta} = \frac{v_L - v_R}{L} \end{cases} \quad (2)$$

where v_L and v_R are the corresponding linear velocities of left and right wheel respectively, and L is measured as the distance

Figure 1. Modeling of a wheel mobile robot.

between two wheels.

3. Navigation Strategies

In the applications of ultrasonic sensors for robot navigation problem, because of the uncertainties caused by the accuracies of sensors, it's hardly for the robot to get the precise distance information. Fortunately, unlike classical logic only permitting propositions having a value of truth or falsity, the fuzzy logic has the ability to produce acceptable but definite output in response to incomplete, ambiguous, distorted or inaccurate input. Thus, by using the fuzzy logic, if the robot can know the approximate values, and then identify the obstacles which are far or near and the situations of danger or not dangerous, this problem might can be solved.

In order to drive the mobile robot real-timely in totally unknown environment, navigation is completely achieved in a reactive manner. The proposed approach here is based on fuzzy inference system and is inspired by human behaviors, which consists of judging the most dangerous direction of coming obstacles while detecting any obstacles (danger judgment strategy). This strategy avoids local minimal by enabling the robot to turn to the safest direction. But some special situations are yet encountered in the case of a wall or concave obstacles. That is why, another strategy, is developed here by coordinating the "danger judgment strategy" and another elementary behavior, of the wall following behavior. When there is no risk of collision between the robot and obstacles include both static and dynamic objects, or the robot has been driven over the obstacles temporarily, another strategy named "target positioning strategy"

is to be activated for seeking the goal position. While the robot moving in the dynamic environment, at the moments of state transition, such as the transitions from "dangerous" to "not dangerous" or from "wall following strategy" to other strategies, in some cases, the robot will fail into the mistake of "infinite repetition" or "dead cycle". Under this circumstance, the robot might keep on moving back, forth, left and right repeatedly. In order to avoid this mistake, a "state memorizing strategy" which is used to record robot's movement state is also developed in this section.

3.1 Danger Judgment Strategy for Obstacle Avoidance

The "danger judgment strategy" developed here is used to avoid convex obstacles or dynamic ones. Firstly, at the start position, the robot moves ahead with the direction towards the target. Once the sensors installed on the robot body detect any objects (obstacles or other robots), this strategy is to be activated. It is obtained by means of a fuzzy logic controller (FLC) and based on a set of rules which generated by human experience.

The sensors installed on the robot body are set to run once by every 0.1 seconds. Thus, in such a short time, we can simplify the relative velocity calculation v_{ro}^i as the following equation to obtain the acceptable approximate solutions:

$$v_{ro}^i \approx \frac{d_{t1}^i - d_{t2}^i}{\Delta t}, \quad (3)$$

where d_{t1}^i and d_{t2}^i are the distance measured by the i_{th} sensor ($i = 0, 1, \dots, 15$) at time $t1$ and $t2$. Here, the time interval Δt of $t1$ and $t2$ is 0.1 seconds. We use the expected time of collision to denote the danger coefficient, that is:

$$R_{dc}^i = d_{t2}^i / v_{ro}^i. \quad (4)$$

The smaller of the value of R_{dc}^i is, the more dangerous the robot will be. Specially, when the direction of vector $\vec{v}_{ro}^i$ points away from the obstacle the value of R_{dc}^i is set as infinity. We denote that the danger coefficients of all the sensors are:

$$\mathbb{R}_{dc} = \{R_{dc}^i | 0 \leq i \leq 15\}. \quad (5)$$

Here, the angle of the corresponding i_{th} sensor with the minimum danger coefficient is judged as the most dangerous direction of coming obstacles, and will be firstly dealt with. The method used for avoiding this dangerous collision is achieved by a fuzzy logic controller which is named as FLC1 in this section. Here, the minimum danger coefficient and the angle of the corresponding sensor are taken as the input variables.

Figure 2. Membership functions for input and output variables of FLC1. S, small; M, medium; B, big; VB, very big; RB, right back; R, right; RF, right front; F, front; LF, left front; L, left; LB, left back; B, back; Z, zero; NB, negative big; NM, negative medium; NS, negative small; PB, positive big; PM, positive medium; PS, positive small; FLC, fuzzy logic controller.

Moreover, the linear velocities of left and right wheel will be the output variables. Then the input variables can be described as:

$$R = \min(\mathbb{R}_{dc}), \quad (6)$$

$$\Phi = \angle s_i, \text{ when } R_{dc}^i = \min(\mathbb{R}_{dc}), \quad (7)$$

where $-180^\circ \leq \angle s_i < 180^\circ$, which is the corresponding angle of the i_{th} sensor. The fuzzification process of a FLC transforms a non-fuzzy input values to a fuzzy value. For fuzzifier, the input and output variables can be divided into several linguistic terms as follow:

S: small; M: medium; B: big; VB: very big;
RB: right back; R: right; RF: right front; F: front;
LF: left front; L: left; LB: left back; B: back;
Z: zero; NB: negative big; NM: negative medium;
NS: negative small; PB: positive big;
PM: positive medium; PS: positive small.

The transformation process of non-fuzzy input values to fuzzy values is achieved by using the membership functions which provide fuzzy terms with a definite meaning. The membership functions for input and output variables of FLC1 are as shown in Figure 2. As shown in this figure, when the main risk comes from left side, angle Φ is positive, otherwise it is negative. Moreover, we use the triangular type of membership functions with the range of $[-1, 1]$ for the output variables,

Table 1. Rules set for FLC1

	v_L	Φ							
		RB	R	RF	F	LF	L	LB	B
R	S	PB	PB	NB	NB	NS	PB	PB	PS
	M	PB	PB	NB	NB	PB	PB	PS	PS
	B	PB	PB	NS	Z	PB	PB	PS	PS
	VB	PB	PB	PS	PS	PB	PB	PS	PS
	v_R	Φ							
		RB	R	RF	F	LF	L	LB	B
R	S	PB	PB	NS	NS	NB	PB	PB	PB
	M	PS	PB	PB	Z	NB	PB	PB	PB
	B	PS	PB	PB	PB	NS	PB	PB	PB
	VB	PS	PB	PB	PB	PS	PB	PB	PB

FLC, fuzzy logic controller; S, small; M, medium; B, big; VB, very big; RB, right back; R, right; RF, right front; F, front; LF, left front; L, left; LB, left back; B, back; Z, zero; NB, negative big; NM, negative medium; NS, negative small; PB, positive big; PM, positive medium; PS, positive small.

which denote the scale of the maximum linear velocity.

The procedure of fuzzy inference usually contains a set of rules which are used to appoint the desired control behavior. A rules set is a condition description taking the form of “IF... THEN...” rules. The rules set applied for FLC1 are as shown in Table 1.

3.2 Wall Following Strategy

In an environment containing a wall or concave obstacles, in order to avoid blocking situation, an additional strategy is used, which consists of following the contour of the obstacle in order to skirt round it. This strategy is achieved by using another fuzzy logic controller (FLC2) where the choice of the rules and the tuning of the parameters are also simply done by human experience.

The goal is to follow the wall surrounding the robot, so the input variable of FLC2 is the angle φ between the moving direction of the robot and the wall. As shown in Figure 3, we assume that the current position of the robot is P_t . The wall now is located on the right side of the robot. The moving direction can be denoted by the dotted arrow OA . In order to reduce the measuring errors, we will use two sets of measured distance which produced by two sets of sensors ($\{s_6, s_8\}$ and $\{s_6, s_{10}\}$) to estimate angle φ . Denote that φ_1 and φ_2 are the

Figure 3. Scheme of wall following.

Figure 4. Membership functions for input variable of FLC2. NB, negative big; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PB, positive big; FLC, fuzzy logic controller.

angle calculated by two sets of sensors respectively, through the geometry knowledge, they can be easily described as:

$$\varphi_1 = \tan^{-1} \left(\frac{(s_6 + r) - (s_8 + r) \times \cos(\pi/8)}{(s_8 + r) \times \sin(\pi/8)} \right) - \frac{\pi}{8}, \quad (8)$$

$$\varphi_2 = \tan^{-1} \left(\frac{(s_6 + r) - (s_{10} + r) \times \cos(\pi/4)}{(s_{10} + r) \times \sin(\pi/4)} \right) - \frac{\pi}{4}, \quad (9)$$

where r is the radius of the robot. Then the included angle φ can be given as the average value:

$$\varphi = (\varphi_1 + \varphi_2)/2. \quad (10)$$

In addition, when the wall is located on the left side of the robot, angle φ can be calculated by using the same method. The output variables of FLC2 also are the velocities of left and right wheel and use the triangular membership functions. The membership functions for the input variable and the set of control rules are given in Figure 4 and Table 2.

In the situation that encountered a concave obstacle, the

Table 2. Rules set for FLC2

φ	NB	NM	NS	Z	PS	PM	PB
v_L	NB	NM	NS	PM	PS	PM	PB
v_R	PB	PM	PS	PM	NS	NM	NB

FLC, fuzzy logic controller; NB, negative big; NM, negative medium; NS, negative small; Z, zero; PS, positive small; PM, positive medium; PB, positive big.

Figure 5. Scheme of target positioning.

robot will make the judgment that which side is much safer in accordance with the distance measurements of all the sensors. At position P_{t+1} shown in Figure 3, the front and right side give small measuring distance, thus, the robot will turn left first, and then the wall following approach will continue to be done.

3.3 Target Positioning Strategy

When there is no risk of collision, another strategy for target positioning is to be active. The new FLC which the target orientation process is achieved by is denoted as FLC3.

The schematic model of target positioning is as given in Figure 5. In this figure, with the current position P_R , the robot has moved over the obstacle P_O . With the angle ψ_t between the line from the center of the robot to the target point and the x-axis, we denote that the angular difference is $\Psi = \theta - \psi_t$. Now, the obstacle is moving far away from the robot and poses no risk to the robot. Thus, the strategy of FLC3 is come into use. This controller is used to drive the robot to point to the target position. In the other words, it is used to reduce the angular difference Ψ which is one of the input variables of FLC3.

Figure 6. Robot navigation in U-valley environment without (a) and with (b) state memorizing behavior.

Another constraint for FLC3 is the distance D between the robot and its target, which is denoted by the multiple of robot's radius in this paper. Moreover, the output variables also are the linear velocities of left and right wheel. Similar with FLC1, the triangular type of membership functions and another experimental rules set are used for both input and output variables of FLC3.

3.4 State Memorizing Behavior

Under some circumstances, the fuzzy behavior-based robot, like other reactive robots, suffers from the “infinite repetition” or “dead cycle” problems. These problems may occur in some cases, even if the wall following behavior has been added to the fuzzy inference system. Thus, designing several mandatory-turn rules when then fails into the symmetric indecision situations and memorizing the moving state are likely became useful. For example, as shown in Figure 6(a), the robot indefinitely wanders in a loop inside a U-valley shape obstacle, because it can't remember the position visited before, and its navigation is only based on the local sensed environment.

In Figure 6(a), firstly, the robot moves directly toward the target position according to FLC3 by using the target positioning behavior. Because there are no obstacles sensed on the way of robot, the robot judges that it is the shortest path to the target. Once the obstacle is detected in front, the robot is guided to turn left constrainedly at position 1 by using the mandatory-turn rules. According to the distance measurements of surround environment, a wall has been found, then, the wall following behavior is activated. After arriving position 2, according to the feedback information from sensors, a concave obstacle has been identified. Because the sensors on the left side of robot give big measured distance, the robot will turn left, and then, continue

moving and following the wall by FLC2 until arrived position 3. After position 3, there are either no obstacles or far away obstacles detected by the robot, so the robot is attracted to the target according to the target positioning behavior. Similarly, the robot follows the path from position 4 to 7 via 5 and 6, and then come back to position 2 via 8 and 9. An “infinite repetition” or “dead cycle” occurs.

The study on the trajectory in Figure 6(a) found that position 3 and 7 are the turning points on the “dead cycle” path. At position 3, if the robot can turn right and go on following the wall instead of turn to towards the target, this problem may can be resolved. Thus, at position 3, if the robot can remember that the concave obstacle just on the right side and the wall following process hasn't been finished, it will turn right and try to pass around this concave obstacle. Based on this idea, a state memorizing strategy is developed here.

Another situation which the robot needs to “remember” the movement states is the “head-on crash” situation for two robots. We assume that there are two robots moving on the opposite way and will fall into danger of head-on crash. According to the danger judgment behavior, the direction of main coming danger is right ahead for both two robots. Then, both two robots will move backwards and the risk of collision disappear. After moving backwards, the risk of collision is zero, then, both robots move ahead again. Obviously, the head-on crash danger comes again, and then both of them move backwards. An “infinite repetition” occurs. Thus, if the robot can remember the moving state of “dangerous” or “not dangerous”, when the states have repeated for several times, the robot will make a decision of moving direction according to the measured distances of sensors, then, this problem may can be resolved.

As discussed above, we denote $M = [m_1, m_2, m_3]$ to memorize the moving state of the robot. The meaning of every parameter is as shown in Table 3.

The flow diagram of the proposed method together with the fuzzy based behaviors is shown in Figure 7. Initially, three parameters of M are initialized to 0. After the robot arriving position 3 in Figure 6(b), both m_1 and m_2 have been changed to 1 at position 1, while m_3 has been changed to 2 at position 2. Because the robot has remembered that there is a concave obstacle and is located on its right side, then, the robot will turn right at the turning points 3 and 4 to move along the contour of this obstacle. After the robot arriving position 5, the measured distances of all the sensors show that it has left far enough from the obstacle, thus, it will adjust the moving direction towards the target.

Table 3. Parameters of state memorizing strategy

Parameter	Value	Meaning
m_1	0	No risk of collision, not dangerous
	1	A risk of collision, dangerous
m_2	0	Wall following strategy end
	1	Wall following strategy start
m_3	0	No concave obstacles
	1	Concave obstacle on robot's left side
	2	Concave obstacle on robot's right side

Figure 7. Flow diagram of proposed algorithm. FLC, fuzzy logic controller.

4. Simulation Experiments

To demonstrate the effectiveness of the proposed navigation algorithm, simulations with different environments are implemented using Matlab. In this section, R_i and T_i are used to denote the i_{th} robot and its corresponding target position. All the robots have the same size and move on a horizontal plane using the exact same navigation strategies. The radius of robots is set as 2.5 unit length while the maximum velocity of left and right wheel also is 2.5 (unit length per second). To denote the movement trajectories, we will record the positions and moving

Figure 8. Result in the environment with convex obstacles.

directions of all the robots by every one second.

4.1 Simulation with Convex Obstacles

A typical experiment in the environment with convex obstacles is as shown in Figure 8. The environment where three robots are plotted in includes both geometric and circular shape of obstacles, meanwhile, all the obstacles are static here. The start position and its corresponding target for every robot also are denoted in this figure. Firstly, the robots move towards the corresponding targets under the control of target positioning behavior until detected any obstacles. After a wall or a much bigger obstacle is identified, the wall following behavior is to be activated. From this figure we can see that the proposed method shows excellent performance for avoiding convex obstacles.

The recorded velocity variation curves of both wheels in this simulation are presented in Figure 9. It can be seen that the velocities of both wheels of all the robots increased when the robots moved towards the targets in the process of wall following; and the velocities of left or right or both wheels decreased on different levels to generate the velocity differences for turning left or right when avoiding obstacles. At the end of the trajectories, the velocities of both wheels were decreased to reach the targets.

Figure 9. Velocity variation curves with convex obstacles.

Figure 10. Results in U-valley environment without (a) and with (b) state memorizing behavior.

4.2 Effectiveness of the State Memorizing Strategy

In order to verify the proposed state memorizing strategy when the robot moves in the indoor environments with concave obstacles, a simulation for a typical complicated situation that causes the “infinite repetition” or “dead cycle” problem has been done in this section.

A robot moving under a U-valley shape obstacle situation is shown in Figure 10. The robot moved in the U-valley area towards the target first, but because of lack of “memory function”, as shown in Figure 10(a) the robot turned back towards the target again and again after finishing wall following process. In contrast, as shown in Figure 10(b), the robot “remembered” the moving state and made a decision to turn right to keep on following the wall, so that it can move around the obstacle and finally reached the target.

4.3 Simulation in the Indoor Environments

The indoor environments contain not only convex obstacles but also concave ones and walls. In this section, the proposed indoor environments are formed by several convex obstacles

Figure 11. Result in the indoor environments.

Figure 12. Velocity variation curves in the indoor environments.

and concave walls, and a dynamic obstacle which denoted by a white circular shape is also added. The trajectories of all the robots are as shown in Figure 11. Here, there are 4 robots that with different start positions and different targets. The size of the environment is set as 400×400 unit length. The dynamic obstacle will keep on moving left and right between $[300, 200]$ and $[100, 200]$ during the whole process at the speed of 0.8 times of maximum linear velocity of wheels, and its trajectory is denoted as a dashed line.

The velocity change curves for all the robots in this simulation are presented in Figure 12. Refer to Figure 11, robot 1 encountered the dynamic obstacle at position A. It can be seen from Figure 12 that, in order to avoid collision, the velocity of left wheel of robot 1 was rapidly decreased to guide the robot to turn left first, and then increased again to drive the robot to move ahead. Similarly, robot 2 and 4 met at position B. Accord-

ing to the velocity change curves we can see that, robot 4 turned left first and then moved ahead at a high speed, meanwhile, robot 2 span to its right side on the spot to let robot 4 to pass through first.

5. Conclusions

This paper studied a real-time reactive navigation problem for multiple mobile robots in the unknown indoor environments. The proposed approach is formed by several behaviors and based on fuzzy inference system. In this paper, both convex and concave obstacles were investigated. Under the control of the proposed method, the robots can autonomously avoid both static and dynamic obstacles and reach the targets. The simulations provided satisfactory results for the proposed problem and the fuzzy-based strategies have been verified to be effective.

In the real applications of the future work, the uncertainties caused by the accuracies of sensors should be considered and the redundancy safe distance should be adopted in the application of the proposed wall following strategy. The performance of a fuzzy inference is influenced by its knowledge base (rules). Thus, it is essential to adjust the rules of fuzzy controller to get a better performance, which will be the next work for us. The adjust method can be viewed as an optimization process, so a neural network or a genetic algorithm offers a possibility to solve this problem.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

References

- [1] D. Janglova, "Neural networks in mobile robot motion," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 15-22, 2004. <http://dx.doi.org/10.5772/5615>
- [2] C. E. Thomas, M. A. C. Pacheco, and M. M. B. R. Vellasco, "Mobile robot path planning using genetic algorithms," in *Foundations and Tools for Neural Modeling*, J. Mira and J. V. Sanchez-Andres, Eds. Berlin: Springer-Verlag, 1999, pp. 671-679. <http://dx.doi.org/10.1007/BFb0098225>
- [3] R. Carelli, C. M. Soria, and B. Morales, "Vision-based tracking control for mobile robots," in *Proceedings of 12th International Conference on Advanced Robotics (ICAR'05)*, Seattle, WA, 2005, pp. 148-152. <http://dx.doi.org/10.1109/ICAR.2005.1507405>
- [4] W. J. Yim and J. B. Park, "Analysis of mobile robot navigation using vector field histogram according to the number of sectors, the robot speed and the width of the path," in *Proceedings of 2014 14th International Conference on Control, Automation and Systems (ICCAS)*, Seoul, Korea, 2014, pp. 1037-1040. <http://dx.doi.org/10.1109/ICCAS.2014.6987943>
- [5] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proceedings of IEEE International Symposium on Industrial Electronics (ISIE)*, Busan, Korea, 2001, pp. 1530-1535. <http://dx.doi.org/10.1109/ISIE.2001.931933>
- [6] R. Simmons, "The curvature-velocity method for local obstacle avoidance," in *Proceedings of IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 1996, pp. 3375-3382. <http://dx.doi.org/10.1109/ROBOT.1996.511023>
- [7] J. P. van den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885-897, 2005. <http://dx.doi.org/10.1109/TRO.2005.851378>
- [8] C. G. Zhang and Y. G. Xi, "Rolling path planning and safety analysis of mobile robot in dynamic uncertain environment," *Control Theory & Applications*, vol. 20, no. 1, pp. 37-44, 2003.
- [9] T. Jin, "Obstacle avoidance of mobile robot based on behavior hierarchy by fuzzy logic," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 12, no. 3, pp. 245-249, 2012. <http://dx.doi.org/10.5391/IJFIS.2012.12.3.245>
- [10] H. G. Nguyen, W. H. Kim, and J. H. Shin, "A study on an adaptive robust fuzzy controller with GAs for path tracking of a wheeled mobile robot," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 10, no. 1, pp. 12-18, 2010. <http://dx.doi.org/10.5391/ijfis.2010.10.1.012>
- [11] D. Zhao and T. Zou, "A finite-time approach to formation control of multiple mobile robots with terminal sliding mode," *International Journal of Systems Science*, vol. 43,

no. 11, pp. 1998-2014, 2012. <http://dx.doi.org/10.1080/00207721.2011.564323>

- [12] X. Zhong, X. Zhong, and X. Peng, "Velocity-Change-Space-based dynamic motion planning for mobile robots navigation," *Neurocomputing*, vol. 143, pp. 153-163, 2014. <http://dx.doi.org/10.1016/j.neucom.2014.06.010>
- [13] T. P. Nascimento, A. P. Moreira, A. G. Scolari Conceicao, and A. Bonarini, "Intelligent state changing applied to multi-robot systems," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 115-124, 2013. <http://dx.doi.org/10.1016/j.robot.2012.10.011>
- [14] D. R. Parhi, S. K. Pradhan, A. K. Panda, and R. K. Behera, "The stable and precise motion control for multiple mobile robots," *Applied Soft Computing*, vol. 9, no. 2, pp. 477-487, 2009. <http://dx.doi.org/10.1016/j.asoc.2008.04.017>
- [15] R. Zhao, D. H. Lee, and H. K. Lee, "Mobile robot navigation using optimized fuzzy controller by genetic algorithm," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 15, no. 1, pp. 12-19, 2015. <http://dx.doi.org/10.5391/IJFIS.2015.15.1.12>

Ran Zhao received his B.S. degrees in Department of Information Engineering from Shandong University, China, in 2006, and the M.S. degree in Department of Electrical and Electronics Engineering from Korea Uni-

versity of Technology and Education, Korea, in 2009. He is currently working toward a Ph.D. degree. His research interests include intelligent mobile robot, fuzzy logic control and genetic algorithm.

E-mail : zhaoran0922@163.com

Dong Hwan Lee received his B.S., M.S., and Ph.D. Candidate degrees in Department of Electrical Engineering from Korea University of Technology and Education, Cheonan, Korea in 1999, 2006, and 2010, respectively.

Since 2006, Department of Smart Grid Electric, Electronic and Control, Korea Polytechnic Colleges, where he is currently a Professor. His major research interests include control theory and intelligent mobile robot, PLC control, control of automation system, and genetic algorithms.

E-mail : k2shine@kopo.ac.kr

Hong Kyu Lee received his B.S., M.S, and Ph.D. degrees in Department of Electronics Engineering from Seoul National University, Seoul, Korea in 1977, 1979, and 1989, respectively. He was with the Agency of Defense

Development from 1979 to 1992. Since 1992, Department of Electrical Engineering, Korea University of Technology and Education, where he is currently a Professor. His major research interests include control theory and intelligent mobile robot, fuzzy logic control, neural networks, and genetic algorithms.

E-mail: hongkyu@koreatech.ac.kr