Mobile Robot Navigation using Optimized Fuzzy Controller by Genetic Algorithm

Ran Zhao¹, Dong Hwan Lee², and Hong Kyu Lee²

Department of Electrical Engineering, Korea University of Technology and Education, Cheonan 330-708, Korea

Ijfis

Abstract

In order to guide the robots move along a collision-free path efficiently and reach the goal position quickly in the unknown multi-obstacle environment, this paper presented the navigation problem of a wheel mobile robot based on proximity sensors by fuzzy logic controller. Then a genetic algorithm was applied to optimize the membership function of input and output variables and the rule base of the fuzzy controller. Here the environment is unknown for the robot and contains various types of obstacles. The robot should detect the surrounding information by its own sensors only. For the special condition of path deadlock problem, a wall following method named angle compensation method was also developed here. The simulation results showed a good performance for navigation problem of mobile robots.

Keywords: Fuzzy logic controller, Genetic algorithm, Path planning, Robot navigation, Wall following

1. Introduction

The navigation problem of mobile robots is one of the most popular issues in robot field. Mobile robots have been used and performed some risky and repetitive tasks for human being, such as space research, factory automation and defense security, which economized the labor force to be engaged in other aspects. In these areas, how to detect the surrounding information and guide the movement of mobile robots has become a primary question. So in an environment with obstacles, a mobile robot should have a fundamental skill of finding a collision-free path from a starting location to a planned target location.

In the past few decades, plenty of researchers have applied their energy to solve the navigation problem of mobile robots. Such as, in [1], the researchers proposed a control rule to find out reasonable target linear and rotational velocities of the mobile robot using PID controllers. Moreover, in [2], the authors used only artificial vision to establish both the robots position and orientation relative to the target using non-linear controllers based on Lyapunov stability analysis. The soft computing methods also have been tried to solve this problem such as fuzzy logic, genetic algorithm (GA), neural network, etc. The artificial neural network (ANN) has the ability to learn the situations of the environment. Many researchers have applied the neural network [3, 4] successfully to develop the methods of the navigation problem of a mobile robot. In [5], GAs together with characteristics has the ability in solving such problems and they can approach for optimum results in environment with dynamic obstacles. The fuzzy logic controller (FLC) for the navigation and obstacle avoidance problem of mobile robots

Received: Jan. 31, 2015
 Revised : Mar. 10, 2015
 Accepted: Mar. 17, 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.

also has been studied by many researchers for years. One of the most important questions is to establish a high-efficiency rule set. The performance of FLC is influenced by its knowledge base (rule set) and the membership functions. Thus, it is indispensable to adjust the rules of fuzzy controller and the membership functions of input and output variables to get a better performance. In [6], a GA was applied to modify the input and output membership functions of the fuzzy controller. Researches in [7] combined the fuzzy logic with both GA and ANN for path planning problem of mobile robot in static environment, and three different algorithms were loaded with different effects. Many other researchers [8] have also applied the GA optimized FLC in this area. But most of them used the regular (circle shape) obstacles for simulation. In the environment with irregular obstacles, these methods might not work properly. Here, we studied the navigation problem of a wheel mobile robot in the unknown environment with both regular and irregular obstacles based on proximity sensors by FLC. Then a GA was applied to optimize both the membership functions and the rules set of the fuzzy controller. In the case of path deadlock problem, a wall following method was also proposed.

The rest of this paper is organized as follows: Section 2 presents the modeling of the robot and kinematic functions. Section 3 describes the FLC. In Section 4, we present an optimization method for FLC by GA. Section 5 presents a strategy of exploring the way forward and a wall following method. In Section 6, we verify the effectiveness of the proposed method by simulation. Finally, conclusions and discussions are included in Section 7.

2. Problem Formulation

In order to obtain the required information of environment for navigation, different kinds of mobile robots with different sensors have been used for this purpose. In this work, we choose a classic wheeled robot as an example for simulation, which has two DC motors for left and right wheel and both wheel has an encoder on each one so that the robot can detect how far each motor has moved by itself. As shown in Figure 1, the robot has nine proximity sensors (ultrasonic sensors or infra-red sensors). Totally three groups of sensors have been installed at the front, left side and right side of the robot. Every group has three sensors, and every sensor has been setup for every 30 degrees. These sensors are used to detect the distance to the surrounding obstacles. We assume that the position of all the obstacles is unknown and the start and goal position are known

Figure 1. Structure of wheel robot.

for the robot.

To simplify the simulation, we put two wheels together to consider as the unicycle. Thus, the kinematic functions can be described as the following equations:

$$\begin{cases} \frac{dx_r}{dt} = v \times \cos \theta \\ \frac{dy_r}{dt} = v \times \sin \theta \\ \frac{d\theta}{dt} = \omega \end{cases} \quad (1)$$

where (x_r, y_r) is the coordinates of robots center point. v and ω are the linear and angular velocities of the robot respectively. θ is the angle between the robots moving direction and x-axis.

3. Design of Fuzzy Logic Controller

Fuzzy logic control is one of the most successful areas in the application of intelligent control. The fuzzy logic control theory has shown excellent performance when the processes are too complex for analysis by traditional mathematical description [9]. It includes three components which are fuzzification, fuzzy inference and defuzzification.

The fuzzification process can be used to transforms a non-fuzzy input value to a fuzzy value. As an example, in this work, there are two input variables:

$$D = \min(s_4, s_5, s_6), \quad (2)$$

$$R = \frac{s_3 + s_4}{2} - \frac{s_6 + s_7}{2}, \quad (3)$$

where D is the distance between the forward obstacles and the robot body, and R is the distance difference between front-left and front-right obstacles. s_i is the distance which detected by the i_{th} sensor on the robot. The rest sensors s_1, s_2, s_8 and s_9 are used in the wall following method and to insure the safety

Table 1. Variable range and linguistic terms

Variables	Range	Linguistic terms
D	[0, 6]	{VS, S, M, B, VB}
R	[-6,6]	{RB, RS, CE, LS, LB}
S_a	[-2,2]	{TRB, TRS, Z, TLS, TLB}
S_v	[0, 1]	{VS, S, M, B, VB}

distance while moving. We denote that the maximum linear velocity is V_m , and the maximum angle of the robot rotates for one time is A_m . S_a and S_v are the scale of A_m and V_m respectively, which will be output variables of FLC.

For fuzzifier, the input and output values which with the corresponding range, can be divided into five linguistic terms as shown in Table 1.

Here, the meaning of every linguistic values respectively are: very small (VS), small (S), middle (M), big (B), very big (VB), right big (RB), right small (RS), center equal (CE), left small (LS), left big (LB), turn right big (TRB), turn right small (TRS), zero (Z), turn left small (TLS) and turn left big (TLB).

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 [10]. In this work, we used the triangular membership functions. For the i_{th} linguistic term A_i , a triangular sharp can be defined by three points: $A_{i,l}$, $A_{i,c}$ and $A_{i,r}$, which means the left, center and right point. According to A_i together with the corresponding input x_i , the output value μ_i can be described as Eq. (4):

$$\mu_i(x) = \begin{cases} \frac{(A_{i,c}-A_{i,r})-(A_{i,c}-x_i)}{A_{i,c}-A_{i,l}}, & A_{i,l} \leq x_i \leq A_{i,c} \\ \frac{(A_{i,r}-A_{i,c})-(x_i-A_{i,c})}{A_{i,r}-A_{i,l}}, & A_{i,c} \leq x_i \leq A_{i,r} \\ 0, & \text{others} \end{cases} \quad (4)$$

The process of fuzzy inference usually uses a set of rules to appoint the desired control behavior. A rule is a condition description take the form of IF...AND...THEN... rules. The rules set used here are as shown in Table 2. Specially, when the distance difference R is zero, the robot is fixedly set to turn left.

4. Optimized FLC by GA

The rules set and membership functions discussed in Section 3 were created all by human experience, which will influence the performance of a FLC. Thus, it is indispensable to adjust them to get a better performance. A GA is employed here to do this work. A GA is a search strategy based on models of evolution

Table 2. Rules set of fuzzy logic controller

D	VS	S	M	B	VB
S_v	VS	S	M	B	VB
R					
S_a	RB	RS	CE	LS	LB
VS	TRB	TRB	TLB	TLB	TLB
S	TRB	TRB	TLB	TLB	TLB
M	TRS	TRS	Z	TLS	TLS
B	TRS	Z	Z	Z	TLS
VB	TRS	Z	Z	Z	TLS

in nature. A GA requires several steps including generating an initial population made up of chromosomes (candidate solutions to the optimization problem), calculating a fitness value for each chromosome, finding the best set of chromosomes that will be used to form the next generation, perform a reproduction selection on the other chromosomes, performing some diversity for the next generation using crossover and mutation, and continuing with the next generation until some stopping criterion is met. Therefore we can achieve the proposed optimization procedure in accordance with the following order.

a) Chromosomes and initialization

A chromosome corresponds to a possible solution of the optimization problem and every chromosome is composed of several encoded genes. In order to encode a FLC, we integrated the proposed encoding procedure with both the membership functions and rules set. We use A_i, B_i, C_i, D_i , ($i = 1, 2, 3, 4, 5$) to denote the five linguistic values of input and output variables. For the triangular membership functions every linguistic value can be described by three points. Hence, we can encode the membership functions which with the length of 60 in the way as shown in Figure 2. In this work, the rules set are formed by the row-based table like Table 2. Each gene expresses one rule which is one cell in the table. The value of each gene is the value of i which is the subscript of the i_{th} linguistic term of the output variables. For example, the rules in the first part of Table 2 can be denoted as the genes shown in Figure 3. Thus, the total length of the integrated chromosome is 90, which is composed of the proposed two parts.

GAs are efficient techniques for searching for global optimum solutions but may have premature convergence problems [11]. In our case, the number of original individuals of the first generation is initialized as 30. In order to improve the convergence rate of GA and avoid the premature convergence,

Figure 2. Encoding for membership functions.

Figure 3. Encoding for rules set.

we added three same individuals which generated by the input and out values and rules set discussed in the previous section. The rest 27 individuals are produced randomly with real number coding format.

b) Fitness function

In our case, a feasible fitness function should include two elements, that is, efficiency and security. Thus, for the i_{th} individual, the running time T_i of the process from the start location to target location, the total path length L_i and the distance d_i between the robot body and the surrounding obstacles, which detected by the robots own sensors, should be taken into account. The path length can be described by the following equation:

$$L_i(t) = \int_0^{T_i} dv \times dt \quad (5)$$

A penalty function is required to insure that the robot will not collision any obstacles. Denote that d_s is the safety distance, the penalty function can be designed as:

$$d_i = \begin{cases} 1, & d_i \leq d_s \\ 0, & d_i > d_s \end{cases} \quad (6)$$

Then the fitness function can be written as:

$$F_i = \frac{1}{w_1 L_i + w_2 T_i + w_3 d_i}, \quad (7)$$

where w_1, w_2, w_3 are the weighting coefficients.

c) GA operators

Figure 4. Evolution of the maximum fitness values.

In the proposed algorithm, we used three traditional GA operators: selection, crossover and mutation. Here the ‘stochastic universal sampling’ (SUS) is used as the selection operator. Moreover, the ‘double-point crossover operator’ and ‘real-valued mutation operator’ were used for crossover and mutation procedure in this work.

By following the above steps, now we can implement the optimization procedure. Several essential parameters are defined as: crossover rate is 0.9, mutation rate is 0.01 and maximum generation is 1300. Therefore, through the evolution, we can get the curves of the changing of the maximum fitness values in every generation as shown in Figure 4.

In Figure 4, we compared three different initialization methods: curve (a) is the method which used 3 experiential individuals generated by the FLC controller discussed in Section 3 and 27 random generated individuals. The result with completely random generated 30 individuals is shown as curve (b). Moreover, curve (c) is generated by completely experiential 30 individuals. From these curves we can see that curve (a) shows the maximum fitness value at the 431th generation, which is the fastest. Because of the lacking of population diversity, curve (c) prematurely fall into local extremum. Hence, these 3 empirically generated individuals can improve the convergence speed and insure the diversity of population at the same time.

The optimized membership functions compared with the original ones are shown in Figure 5. Table 3 describes the evolved rules set from Table 2.

5. Obstacle Avoidance and Wall-Following Method

Our purpose is to guide the mobile robot with a collision-free path to move in multi-obstacle environment and reach the target position successfully. Firstly, the robot will rotate towards

Figure 5. Original and optimized membership functions.

Table 3. Evolved fuzzy rules set

D		VS	S	M	B	VB
S_v		S	VS	S	M	B
R						
D	S_a	RB	RS	CE	LS	LB
	VS	TRB	TRS	TLB	TLB	TLB
	S	TRB	TRB	Z	TRS	Z
	M	TRS	Z	Z	TLS	TLB
	B	Z	TRB	TRS	TLS	TRS
	VB	TRS	TRS	TRS	Z	TLB

the target and move ahead with the maximum speed. Once the sensors detect any obstacle block the way, a FLC is to be activated. Figure 6 described the flowchart of the process of obstacle avoidance using the proposed algorithm. Specially, if there is a wall on the way of robot or a u-valley deadlock condition, the wall following method which will be discussed here will be used to find a way to escapes out.

Refer to the coordinates system shown in Figure 1, at time t , according to the output values of FLC we can get the rotation angle α_t and the velocity v_t as:

$$\alpha_t = S_a(t) \times A_m, \quad (8)$$

$$v_t = S_v(t) \times V_m, \quad (9)$$

Figure 6. Flowchart of the proposed algorithm. GA: genetic algorithm.

where $S_a(t)$ and $S_v(t)$ are the output values of the FLC at time t . Then, we can describe the kinematic functions as:

$$\begin{cases} \frac{dx_r(t)}{dt} = v_t \times \cos \theta \times \cos \alpha_t \\ \frac{dy_r(t)}{dt} = v_t \times \sin \theta \times \sin \alpha_t \\ \frac{d\theta(t)}{dt} = \omega_t \end{cases} \quad (10)$$

In order to ensure safety, before the robot moves ahead, firstly the robot will check whether the distance between the robot body and obstacles is far enough than the safety distance. As shown in Figure 7, the robot has moved from the position P_{t-1} to the current position P_t . The robot now is too close to the obstacle. Instead of moving ahead immediately, the robot will temporarily stop and rotate with a certain small angle ϕ toward the side, which is the more safety direction and detected by the sensors on both sides. Then detect the distance again. If the distance now is also smaller than the safety distance, the robot will keep on rotating with the angle ϕ again and again until achieves a safe direction. Assume that after k times of rotation the robot got a safe direction and moved to the next position P_{t+1} . In this procedure, the rotation angle should be:

$$\alpha_t = k \times \phi. \quad (11)$$

In the special case with big obstacles (or a wall) or a deadlock condition (e.g., u-valley obstacle), the robot might be guide further and further from the goal position or moves back and forth inside the u-valley obstacle. Thus, if the robot can move

Figure 7. Process of exploring the way forward.

Figure 8. Wall following method.

along the edge of the obstacle until leave the block area, it will be able to escape the problem. As shown in Figure 8, a wall following method named angle compensation method was developed here.

As shown in Figure 8, we assume that the current position of the robot is P_t . The obstacle is now on the right side of the robot. The current direction can be denoted by the dotted arrow OA . OB and OC are the direction of 8_{th} and 9_{th} sensor respectively. Thus, according to this figure, if the robot can rotate right with the angle φ , the moving direction will parallel to the edge of the obstacle. Through the geometry knowledge, the angle φ can be easily described as:

$$\varphi = \tan^{-1} \left(\frac{(s_7 + r) - (s_8 + r) \times \cos(30^\circ)}{(s_8 + r) \times \sin(30^\circ)} \right) - 30^\circ, \quad (12)$$

where r is the radius of the mobile robot.

6. Simulation Results

In order to verify the effectiveness of the algorithm we have discussed, a series of simulations have been implemented with Matlab. The multi-obstacle environment was set as 100×100 unit-length, and the start and goal position are $(0, 0)$ and $(100, 100)$ respectively. The accelerated speed wont be considered, and the radius and the maximum speed of the robot are set as 1.5 unit-length and 4 unit-length per second. We will record the position of the robot by every 0.5 second. Both the regular

Figure 9. The path before (left) and after (right) evolution with circle shape obstacles.

Figure 10. The velocity (up) and turning angle (down) variation curve before and after evolution with circle shape obstacles.

(circle shape) and irregular (geometrical shape) obstacles have been simulated in this work.

The results in the environment with regular obstacles are shown in Figures 9 and 10. In this case, we only consider the periphery of each obstacle, which can be described by a circle shape. As shown in Figure 9, the obstacles (circle) are much bigger than the robots radius. Thus, when the robot is too close to the obstacles the proposed wall following method will be activated. Compare the path generated by evolved and un-evolved FLC, we can find that the evolved FLC demonstrated almost useless to shorten the path length. However, the running time has been observably decreased from 47.7938 to 41.7437 seconds. Compared with the results of performance gain (7.63% in running time) in [6] and that (1.72% in path length) in [8], the method proposed in this work can get the 12.66% of performance improvement, which should be more effective. Similarly, in Figure 10, the amplitude variation of turning angle over the whole path was not obvious before and after evolution. Nevertheless, the velocity was increased.

In order to verify our analysis, a series of simulations have been achieved with different numbers (form 2 to 25) of circle shape obstacles (see Figure 11). From this figure we can see that

Figure 11. Variation curve of running time (up) and path length (down) in environment with different number of circle sharp obstacles.

Figure 12. The path before (left) and after (right) evolution with geometrical shape obstacles.

Figure 13. The velocity (up) and turning angle (down) variation curve before and after evolution with geometrical shape obstacles.

no matter how many obstacles there are the running time has obviously reduced by the optimized FLC. These simulations, out the other side, proved that the proposed wall following method demonstrated excellent performance in obstacle avoidance and smoothness of the path.

The simulation results with irregular obstacles are shown in Figures 12 and 13. In Figure 12, the simulation with several irregular obstacles composed by different kinds of geometrical shapes was performed. Here, the running time of the robot is

decreased from 55.1929 to 48.8116 seconds.

7. Conclusions

This paper studied a navigation problem with path planning for the mobile robot in unknown environment using FLC and GA. The information of environment and the distance between the robot body and obstacles were detected all by the robots own sensors. A GA was used to optimize the membership functions and rules set of FLC. The simulation results showed that the FLC and proposed wall following method have excellent performance in path planning and obstacle avoidance process. Compare the paths generated by the optimized and un-optimized fuzzy controller the running time of the navigation procedure was observably decreased.

In this work, all the obstacles were fixed with certain position. Future research will focus on extending the algorithm to an environment with dynamic obstacles. In addition, more experiments can be done by applying the proposed method to a real robot.

Conflict of Interest

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

References

- [1] P. K. Padhy, T. Sasaki, S. Nakamura, and H. Hashimoto, "Modeling and position control of mobile robot," in *Proceedings of 11th IEEE International Workshop on Advanced Motion Control*, Nagaoka, Japan, 2010, pp. 100-105. <http://dx.doi.org/10.1109/AMC.2010.5464018>
- [2] 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>
- [3] D. Janglová, "Neural networks in mobile robot motion," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 15-22, 2004.
- [4] K. H. Low, W. K. Leow, and M. H. Ang Jr, "Integrated planning and control of mobile robot with self-organizing neural network," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'02)*, Washington, DC,

2002, pp. 3870-3875. <http://dx.doi.org/10.1109/ROBOT.2002.1014324>

- [5] C. E. Thomaz, M. A. C. Pacheco, and M. M. B. Vellasco, "Mobile robot path planning using genetic algorithms," in *Foundations and Tools for Neural Modeling*, J. Mira and J. V. Sanchez-Andrés, Eds. Berlin: Springer-Verlag, 1999, pp. 671-679. <http://dx.doi.org/10.1007/BFb0098225>
- [6] I. Hassanzadeh and S. M. Sadigh, "Path planning for a mobile robot using fuzzy logic controller tuned by GA," in *Proceedings of 6th International Symposium on Mechatronics and its Applications (ISMA'09)*, Sharjah, UAE, 2009, pp. 1-5. <http://dx.doi.org/10.1109/ISMA.2009.5164798>
- [7] T. Khelchandra, J. Huang, and S. Debnath, "Path planning of mobile robot with neuro-genetic-fuzzy technique in static environment," *International Journal of Hybrid Intelligent Systems*, vol. 11, no. 2, pp. 71-80, 2014. <http://dx.doi.org/10.3233/HIS-130184>
- [8] Q. Liu, Y. G. Lu, and C. X. Xie, "Optimal genetic fuzzy obstacle avoidance controller of autonomous mobile robot based on ultrasonic sensors," in *Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO'06)*, Kunming, China, 2006, pp. 125-129. <http://dx.doi.org/10.1109/ROBIO.2006.340327>
- [9] G. B. Lee and Y. S. Kim, "An image contrast enhancement technique using the improved integrated adaptive fuzzy clustering model," *Journal of the Korean Institute of Intelligent Systems*, vol. 11, no. 9, pp. 777-781, 2001.
- [10] "Introduction to Fuzzy Systems," in *Proceedings of Electronic Technology Directions to the Year 2000*, Adelaide, Australia, 1995, pp. 94-103. <http://dx.doi.org/10.1109/ETD.1995.403485>
- [11] H. K. Lee, D. H. Lee, R. Zhao, G. K. Lee, and M. Lee, "On parameter selection for reducing premature convergence of genetic algorithm," in *Proceedings of the ISCA 23rd International Conference on Computer and Their Applications*

in *Industry and Engineering (CAINE2010)*, Las Vegas, NV, 2010, pp. 214-219.

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 University 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