

Modeling of vision based robot formation control using fuzzy logic controller and extended Kalman filter

Angga Rusdinar and Sungshin Kim *

Electrical Engineering Department, Pusan National University, Busan, 609-735, Korea

Abstract

A modeling of vision based robot formation control system using fuzzy logic controller and extended Kalman filter is presented in this paper. The main problems affecting formation controls using fuzzy logic controller and vision based robots are: a robot's position in a formation need to be maintained, how to develop the membership function in order to obtain the optimal fuzzy system control that has the ability to do the formation control and the noise coming from camera process changes the position of references view. In order to handle these problems, we propose a fuzzy logic controller system equipped with a dynamic output membership function that controls the speed of the robot wheels to handle the maintenance position in formation. The output membership function changes over time based on changes in input at time $t-1$ to t . The noises appearing in image processing change the virtual target point positions are handled by Extended Kalman filter. The virtual target positions are established in order to define the formations. The virtual target point positions can be changed at any time in accordance with the desired formation. These algorithms have been validated through simulation. The simulations confirm that the follower robots reach their target point in a short time and are able to maintain their position in the formation although the noises change the target point positions.

Keywords : Formation control, Fuzzy logic controller, Extended Kalman filter

1. Introduction

The formation control problem in robots has been widely studied over the past several years. We can safely say that the formation control requires a multi-agent system. A multi-agent system is a system that consists of more than one agent, i.e. vehicle that employs several sensor/actuators and has the capability to communicate with other agents in order to perform coordinates tasks.

Several methods have been proposed to solve the formation control problem and its applications. Jennings et al. [1] used formation control for search and rescue applications. They employed MOVER, a programming system used for distributed tasks and state communications. Desai et al. [2] used graph theory to maintain a desired formation and to change the formation when required. They modeled the team in three parts (g, r, H), consisting of a group element g that describes the gross position of the lead robot. r describes the relative positions of the robots and H describes the behavior of the robots in the formation.

Uncertainty and imprecision are the two main problems encountered

in a control system. Errors arising from imprecision are often associated with measurement. Several techniques have been proposed to solve the control problems. A Fuzzy Logic Controller (FLC) is well suited to handle these problems. Fuzzy logic has become a means of collecting empirical knowledge and experience and then deal with the uncertainties in the control process. The fuzzy logic system introduced by Zadeh [3] has become a popular topic in control engineering because it is considered by designers to be the simplest solution available for a specific problem. The advantages of fuzzy logic over more traditional solutions is that it allows computers to reason like a human, responding effectively to complex inputs in order to deal with linguistic control situation.

Membership functions and rules are two importance aspects of FLCs. Many techniques have been introduced in order to develop a membership function and its rules. Jou et al. [4] have presented a type of adaptive fuzzy logic controller (AFLC). Their system can perform an adaptive fuzzy inference process using various inference parameters, such as shape and location dynamically and quickly.

This paper presents formation control algorithms using a fuzzy logic controller system (FLC) employing dynamic membership functions. Through the use of these dynamic membership functions, the robots have the ability to control their speed and maintain their formation position. We used FLC because fuzzy logic is well suited to low cost implementations based on cheap sensors. Such systems can be

Manuscript received May. 2, 2012; revised Aug. 12, 2012; accepted Sep. 10, 2012

* Corresponding Author : sskim@pusan.ac.kr

"This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Special Environment Navigation /Localization National Robotics Research Center support program supervised by the NIPA(National IT Industry Promotion Agency)." (NIPA-2012-H1502-12-1002)

© The Korean Institute of Intelligent Systems. All rights reserved.

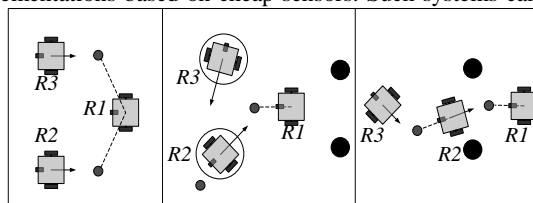


Fig. 1. The robots in triangular and line formations.

easily upgraded by adding new rules that improve performance or address new features. The algorithm and the simulation were built using Matlab. The robots use a two wheel drive robot system. The speeds of the right and left wheels are controlled by the fuzzy system. The robots use a pan-tilt camera as a sensor to detect the other robot positions and orientations. In each robot, a unique marker is attached to robot body. These markers provide identity information that distinguish one robot from another. Through image processing, not only the robot can be identified but also the angle and distance between the followers and the leader can be calculated. We presented this “landmark” system in a previous paper [5]. Other piece of equipment used by the robots in our simulations is a communication system. The leader has ability to send code to the followers. The code is used by the followers to change the formation based on the formation virtual target point. There are two formations used in our simulation: triangular and line formations.

The main points of this study are:

- i) A fuzzy logic controller system that employs a dynamic output membership function.
- ii) Noises that always appear in vision system are simulated in this study, Extended Kalman filter is used to estimate the target point position.

The rest of the paper is organized as follows: The formation control system model is described in Section 2. The Extended Kalman filter is described in section 3. The Fuzzy logic control system for the robot controls is explained in Section 4. The simulation results are examined in Section 5. Our conclusions and possible future improvements are discussed in Section 6.

2. The formation control system model

The distance, the leader robot orientation, and the leader position angle estimated by the follower robots are used as the basic variables in the formations control system. The formation is determined using virtual points that are calculated from the leader position. The virtual points therefore are the displacement from the leader position. In this study, the variables are calculated according to the followers frame, or in other words, the followers position in the global frame is $P_{xy}(0,0)$.

Fig.1 shows the robots do the formation in triangular and line formations. The formation system starts with the triangular formation. The followers maintain the formation with respect to the leader. Under this condition, the linear velocity of the follower robots is not constant, because when the leader turns right or left the radii of their respective trajectories are different, so the followers need to control their position in respect to their distance and position in the formation. When the formation is changed from the triangular to line formation, the followers change their virtual point based on the leader position. The leaders are determined by particular identity of the robots. The smaller identity leads the higher identity. In other words, R1 is the leader of R2 and R2 is the leader of R3. Collision avoidance is used when the formation is changed.

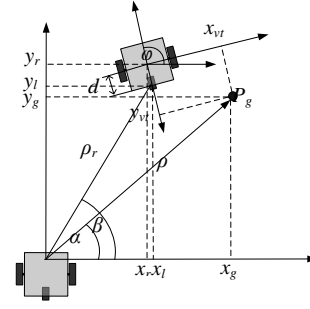


Fig. 2. The coordinate system of leader follower robot.

2.1. The motion models and the coordinate point

Fig. 2 shows the coordinate system used in the robot's formation position. The variables recognized by the sensor or camera are the distance between the follower and leader, the leader orientation, and the angle from follower to leader. The variables are labeled ρ_r , φ , and β respectively. Using these variables, the attached landmark position $P_l(x,y)$ in follower frame is defined by $x_l = \rho_r \sin \beta$ and $y_l = \rho_r \cos \beta$, which can then be written as $P_l = [x_l, y_l + d, \varphi]^T$. The follower robot position is defined by,

$$P_r = R_{z,\varphi} P_l, \quad (1)$$

where $P_r = [x_r, y_r, 0]$ is the leader position in the follower frame and d is distance of the landmark to center point. The virtual target point follower frame is defined by

$$P_g = P_r + R_{z,\varphi} P_{vt}, \quad (2)$$

where $P_{vt} = [x_{vt}, y_{vt}, 0]^T$ is the virtual point position in leader frame, $P_g = [x_g, y_g, \varphi]$ is the virtual point position in follower frame, and $R_{z,\varphi}$ is rotation matrix in z -axis defined by

$$R_{z,\varphi} = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

The variables used in the formation control system are: the error distance between the followers and the virtual points, the error angle view of the follower robot to the virtual points, and the changes in the distance, namely ρ , α , and $\Delta \rho$ respectively. The variables are defined by:

$$\alpha = \arctan 2(x_g, y_g), \quad (4)$$

$$\rho = (x_g^2 + y_g^2)^{1/2}, \quad (5)$$

$$\Delta \rho = \rho_t - \rho_{t-1}, \quad (6)$$

The individual robot kinematic can be represented in following form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos \theta_{t-1} & 0 \\ \sin \theta_{t-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_t \\ \omega_t \end{bmatrix}, \quad (7)$$

where (x, y) represent the position of each robot and θ is the

robot orientation with respect to the global frame. The control parameters for the robot motion are left and right wheel speed, the robot speed and angular velocity, which can be defined by:

$$v = \frac{v_L + v_R}{2}, \tag{8}$$

$$\omega = \frac{v_L - v_R}{L}, \tag{9}$$

where v_L and v_R are the linier velocity of the left and right wheels of the robot, and L is the width of the robot. Since v_L and v_R are the control variables used in the simulation, (7)-(9) can be simplified to:

$$\begin{bmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{\theta}_t \end{bmatrix} = \begin{bmatrix} \cos \theta_{t-1}/2 & \cos \theta_{t-1}/2 \\ \sin \theta_{t-1}/2 & \sin \theta_{t-1}/2 \\ 1/L & -1/L \end{bmatrix} \begin{bmatrix} vL \\ vR \end{bmatrix}. \tag{10}$$

2.2. Collision avoidance scheme

A collision avoidance control algorithm is proposed that uses the partition region algorithm or a partition zone. Fig. 3 displays the partition zone control method. The virtual target points in each zone are determined based on the follower position in the leader robot frame.

There are three zones built for use in the collision avoidance control algorithm. The zones are determined by coordinate from each robots view and the center coordinate is in the robot its self. P_{xy} is the other robot position in the one robot view frame. Zone I is determined by $P_x > 0$ AND $P_y > 0$. Zone II is determined by $P_x < 0$ AND $P_y > 0$. Zone III is determined by $P_y < 0$. P_{xy} is defined by:

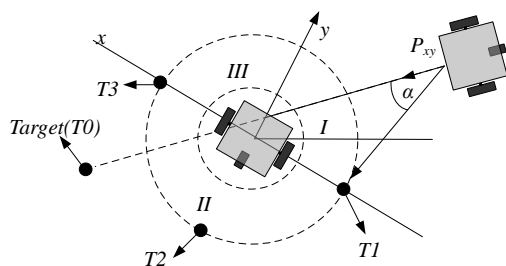


Fig. 3 The partition zone in collision avoidance.

$$P_x = -x_r \cos \varphi - y_r \sin \varphi, \tag{11}$$

$$P_y = x_r \sin \varphi - y_r \cos \varphi. \tag{12}$$

The target selection is defined by

$$T = \begin{cases} T1, & \text{if } (P_x > 0) \wedge (P_y > 0) \\ T2, & \text{if } (P_x < 0) \wedge (P_y > 0) \\ T3, & \text{if } (P_y < 0) \wedge (T0 > T3) \\ T0, & \text{if } (P_y < 0) \wedge (T0 < T3) \end{cases}, \tag{13}$$

where T is target point in each region determined by $T1$, $T2$, and $T3$. $T1$, $T2$, and $T3$ are the virtual target points when the follower robot position is in regions 1, 2, and 3, respectively. The target points are defined using (4), where $T1$, $T2$, and $T3$ are defined by P_g with $P_{vt} = [x_{vt}, 0, 0]^T$, $P_{vt} = [-x_{vt}, 0, 0]^T$, and $P_{vt} = [0, -y_{vt}, 0]^T$ respectively.

3. The Extended Kalman filter

Extended Kalman filter is used to estimate the target point position based on its input (vision) and output (wheels speed). The Kalman filter state is modeled by

$$\bar{x}_t = A x_{t-1} + B u_t + w_{t-1}, \tag{14}$$

and the measurement is modeled as,

$$z_t = H x_t + v_t, \tag{15}$$

where x_t and x_{t-1} are state vector at time t and $t-1$ respectively, A is an $n \times n$ matrix which is related to the state at time $t-1$. B is the $n \times m$ matrix that relate to the control input u , and w_{t-1} is a process noise in time step $t-1$. z_t is a measurement state. H is an $n \times m$ matrix that relate to the measurement z_t , and v_t is measurement noise. In simulation the state is defined as $x_{t-1} = [x_{t-1}, y_{t-1}, \theta_{t-1}]^T$.

Since the movement equation (11) is used, the model of our Extended Kalman filter is simple. Matrix A become identity matrix and H constant became $H=[1 \ 1 \ 0]$ and $B u_t = [\dot{x}_t, \dot{y}_t, \dot{\theta}_t]^T$ where $u_t = [vL, vR]^T$, and B is defined by,

$$B = \begin{bmatrix} \cos \theta_{t-1}/2 & \cos \theta_{t-1}/2 \\ \sin \theta_{t-1}/2 & \sin \theta_{t-1}/2 \\ 1/L & -1/L \end{bmatrix}. \tag{16}$$

Random noises are also added into w_{t-1} and v_t .

The Kalman filter processes consisted of two stages: the time update equation and measurement update equation. The prediction state is given by,

$$x_k = f(x_{k-1}, u_k, 0), \tag{17}$$

$$\bar{P}_t = A_t P_{t-1} A_t^T + W_t E_{x-t} W_t^T, \tag{18}$$

where P_t and \bar{P}_t are the a priori and posteriori estimate of error covariance respectively, E_{x-t} is an process noise matrix

respect to x_t and process noise w_{t-1} , W_t is the Jacobian matrix of partial derivative of f with respect to driving function u , given as $W_t = B$.

The measurement update equations are given by,

$$K_t = \bar{P}_t H^T (H \bar{P}_t H^T + V E_z V^T)^{-1}, \quad (19)$$

$$x_t = \bar{x}_t + K_t (z_t - H \bar{x}_t), \quad (20)$$

$$P_t = (I - K_t H) \bar{P}_t, \quad (21)$$

where K is Kalman gain, E_z is an measurement noise matrix respect to v_t , V is the measurement Jacobian at time t , and I is an identity matrix.

4. The Fuzzy logic controller

The fuzzy logic control system consists of two parallel fuzzy inference systems, FIS-1 and FIS-2. Fig. 4 shows the proposed fuzzy logic controller system. The FIS-1 controls the speed of the right and left robot wheels according to the error distance and error point view follower to leader at time t . The FIS-2 controls the output membership function of FIS-1 according to the error distance and the changes of error distance from prior error distance (at time $t-1$) to the current error distance.

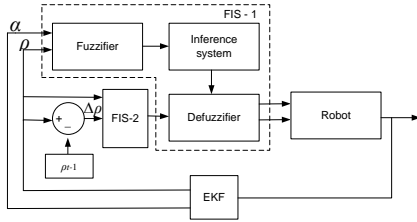


Fig. 4 The proposed fuzzy logic controller system block diagram.

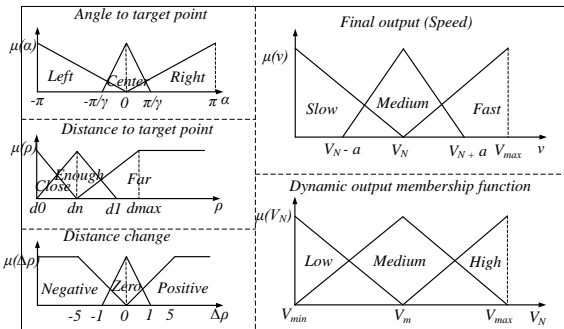


Fig. 5. The membership function in the adaptive fuzzy control system.

4.1 The two wheels drive robot fuzzy control

The inputs entered into the fuzzy logic control system are ρ , α , and $\Delta\rho$ taken from (4)-(6). Fig. 5 shows the all of membership functions in the adaptive control system. There are three inputs membership function ($\mu(\alpha)$, $\mu(\rho)$, and $\mu(\Delta\rho)$) and two outputs

membership function ($\mu(v)$ and $\mu(V_N)$). The error angle fuzzy sets (α) are defined by “Left”, “Center”, and “Right”. They denote that the follower heading is too far to the left, right, or centered on the target point, respectively. γ is the variable used to define the tolerance error of the follower heading and π is the angle in radian.

The fuzzy set for the error distance (ρ) is defined as “Close”, “Enough”, and “Far”. The variables $d0$, dn , $d1$, and $dmax$ are the minimum distance error, the distance error tolerance, the maximum distance error tolerance, and the maximum distance error, respectively. The outputs of the FIS-1 are variables denoted the level of the left and right wheel speeds. The membership function output is dynamic controlled by V_N . The change of V_N is determined by FIS-2 with respect to the changes in the error distance ($\Delta\rho$) and the actual error distance (ρ). The linguistic variable of the fuzzy sets of the output membership functions are defined by “Slow”, “Medium”, and “Fast”.

Table1. The rules of FIS-1

	Left wheel rules			Right wheel rules		
	Left	Center	Right	Left	Center	Right
Close	S	S	S	S	S	S
Enough	M	M	S	S	M	M
Far	F	F	M	M	F	F

The intersection of the error in the robot orientations and the error distance with the respective membership functions $\mu(\alpha)$ and $\mu(\rho)$ is the fuzzy set $\mu(v)$, written as

$$\mu_v = \text{Min}[u_\alpha, \mu_\rho]. \quad (22)$$

There are 18 rules for the two wheel combination. The base rules for the fuzzy system FIS-1 are listed in table 1. Using these, the final fuzzy output in the speed of the right and left wheel are determined using :

$$\text{Final_output} = \frac{\sum_{i=1}^n (\mu_i \times z_i)}{\sum_{i=1}^n \mu_i}, \quad (23)$$

where μ_i is the aggregated output memberships function, and z_i is the output result for every rules, where a (in Fig. 5) is a variable used to define the width of the medium membership function.

Table 2. The rules of FIS-2

	Close	Enough	Far
Positive	M	H	H
Zero	M	M	M
Negative	L	L	M

4.2 The Dynamic output membership function

The error distance changes and the actual error distance are used as inputs in the dynamic membership function. FIS-2 serves to control the output membership function V_N . The linguistic variables of the error distance changes ($\Delta\rho$) are

determined by “Positive”, “Zero”, and “Negative”. The membership function of dynamic V_N is shown in Fig.5. The linguistic variables of the V_N value are determined by “High”, “Middle”, and “Low”. The base rules for FIS-2 listed in table 2.

5. Simulation result

We designed the robots motion simulation and tested the algorithm into the simulation. In simulation, the communication system between robots are assumed already exist and work properly. We simulate that the communication of follower robots only have ability receive data from leader, the leader has ability send data to all followers. From the simulation results, the comparison between dynamic membership function and the static output membership function are showed. Fig. 6 and Fig.7 show the results of the simulation using the static output membership function with two robots. With $V_N=1.5$ for Fig.6 and $V_N=2$ for Fig.7, $a = 0.5$, and $\gamma = 15$ for both of them, Fig. 6.(a) shows that the follower robot was left behind of the leader when the leader turn right. The trajectory indicated by the arrow shows that the trajectory of the follower is close to the leader trajectory. This means that when the follower tried to catch up to the leader, it took the shortest path. This happens because of the speed of the virtual point is higher than the leader speed. Fig. 6.(b) shows the error position of the follower to the target. In the time span from about second 15 to second 20, it can be seen that the error distance between the follower positions to the virtual point became longer because the follower was left behind of the virtual position. In Fig.6.(c), the arrow shows that the position

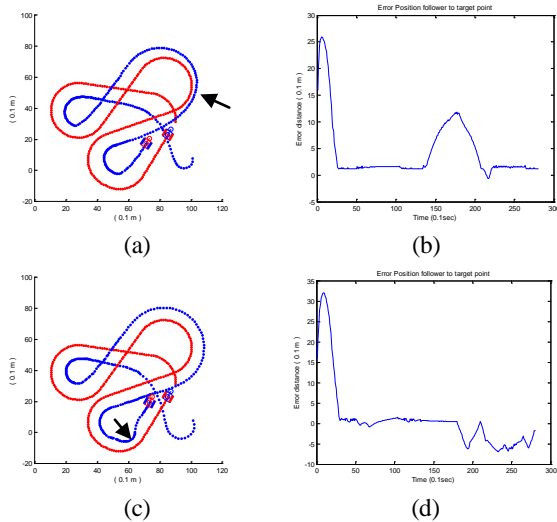


Fig. 6. The robot formation control using static membership function with $V_n=1.5$ (a-b) and $V_n=2$ (c-d).

of the left side follower is too close to the leader, when the leader turns left. It means that follower speed is too high. It makes the radius turning of follower is too long. Fig. 6.(d) shows the error position of follower to the target. It shows that at second 18, the follower position is too close to the virtual

point. This happens because the follower cannot adequately control its speed when the leader moves slowly and turns left.

Fig. 7 shows the trajectory of leader follower robots when the follower used the dynamic output membership functions. It shows that the follower was able to maintain its distance to the target point. This proves that the follower has the ability to control its speed and maintain its position in the formation. Fig. 7.(b) shows the error position of the follower to the target point. It shows that the error is better than when the fuzzy used the static output membership function.

Fig.8 shows the output of V_N value changes every time. The values of the membership function V_N (Using $V_{n_{min}} = 1.5$, $V_{n_{max}} = 2$, and $V_m = 1.7$) are put into the dynamic input membership function fuzzy set (Fig. 5). The V_N was then determined by using the fuzzy rules from table 2. Comparing Fig. 6 to Fig.7, it can be observed that the errors seen at second 15 to 20 were eliminated. This is attributed to the membership function V_N which adapts to the changes in the distance error.

Fig. 9 shows the target point position and target point position with noise. In the figure, the noises from camera processing make the targets point position of follower were not in the actual position. Fig. 10 shows the target point position estimation and target point position with noise. It shows that the EKF able to estimates the target point positions. The black dots indicate the target point read by robot, the green line in Fig. 9 denoted the actual target point, and the red line in Fig. 10 is the estimation position estimated by EKF.

Fig. 11 shows the robots formation control movement. The formation consisted of three robots. The red circles designate

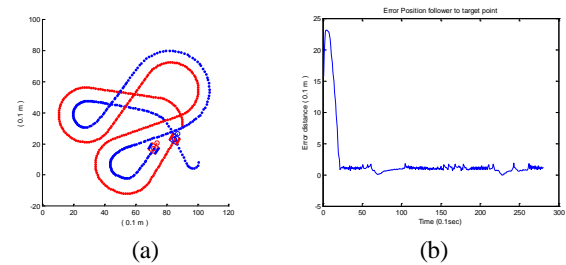


Fig. 7. The robot formation control using dynamic membership function.

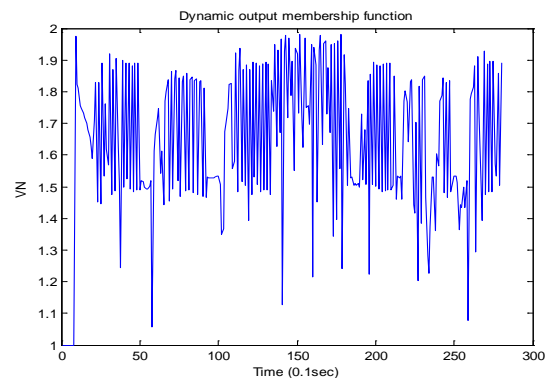


Fig. 8. The membership function in the adaptive fuzzy control system.

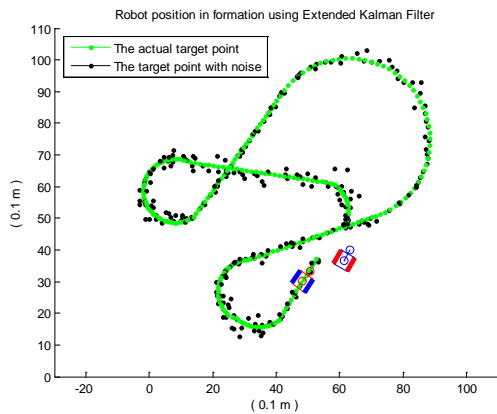


Fig. 9. The actual target point and target point with noise.

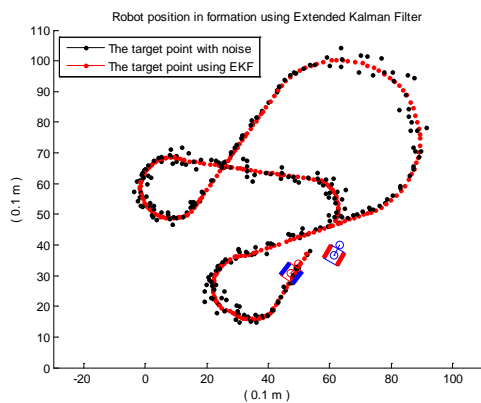


Fig. 10. The target point estimation using EKF and target point with noise.

obstacles. In this simulation the leader is the only robot that has the ability to recognize these obstacles. When the leader detects an obstacle, it sends a code to other robots to change the formation into a line formation. The second robot (R2) follows the leader and the third robot (R3) follows the second robot (R2). Fig. 11(a) shows that the robots are able to maintain the distance and speed necessary to perform a triangle formation and preparing for changing the formation. R2 changes the target point to be in line with the leader. R3 then puts the target point in line with the R2. It is show that R3 is able to avoid a collision with R2 and then follow in line with R2. Fig. 11(b) shows the line formation to avoid the obstacles. Figs. 11(c-d) show that the formation is changed from line to triangular form.

6. Conclusion

This paper presents a formation control method based on a fuzzy logic controller system that employs a dynamic output membership function and Extended Kalman filter. By using this dynamic output membership function, the simulations show that the robots are able to maintain their distance and speed, and are able to form formations according to environmental conditions

with simple rules (table 1 and table 2).

In noise reduction process, Extended Kalman filter has ability

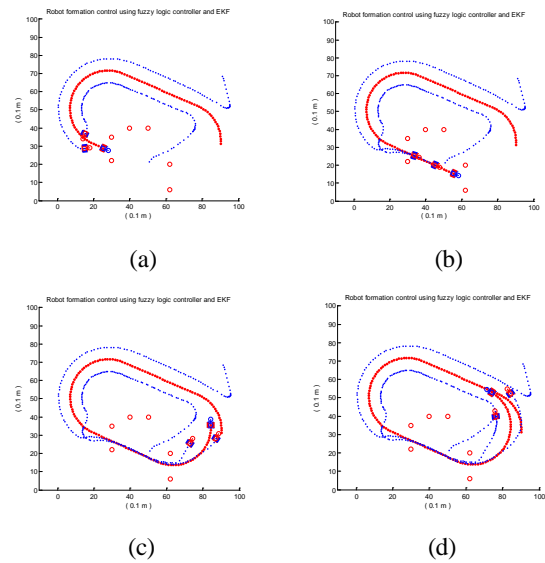


Fig. 11. The membership function in the adaptive fuzzy control system.

to estimate the target point position.

The simulation results show that the proposed algorithm has the ability to form and change the robot formation and avoid collisions between the robots. The simulations movie can be seen in <http://youtu.be/zCASF4I12rA>

References

- [1] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," *Proc. International Conference on Advanced Robotics*, Monterey, CA USA, pp.193-200, 1997.
- [2] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formation of nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol.17, no. 6, pp.905-908, 2001.
- [3] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol.8, pp.338-353, 1965.
- [4] J. M. Jou, P.-Y. Chen, and S. F. Yang, "An adaptive fuzzy logic controller : Its VLSI architecture and applications," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.8, no. 1, pp. 52-60, 2000.
- [5] A. Rusdinar, J. Kim, J. Lee, and S. Kim, "Implementation of real time positioning system using extended Kalman filter and artificial landmark on ceiling," *Journal of Mechanical Science and Technology*, vol.26, no. 3, pp. 949-958, 2012.
- [6] G. L. Mariottini, F. Morbidi, D. Prattichizzo, N. V. Valk, N. Michael, G. Pappas, and K. Daniilidis, "Vision based localization for leader follower formation control," *IEEE Transaction on Robotics*, vol.25, no.6, pp.1432-1438. 2009.

- [7] A. K. Das, R. Fierro, V. Kumar, J. P. Ostrowski, J. Spletzer, and C. J. Taylor, "A vision based formation control framework," *IEEE Transaction on Robotics and Automation*, vol.18, no. 5, pp.813-825, 2002.
- [8] D. Simon, "Sum normal optimization of fuzzy membership function," *International Journal of Uncertainty, Fuzziness, and knowledge based system*, vol.10, no.4, pp.363-384, 2002.
- [9] C. C. Lee, "Fuzzy logic in control systems : Fuzzy logic controller – Part 1," *IEEE Transactions on System, Man, and Cybernetics*, vol.20, no. 2, pp. 404-418, 1990.
-



Sungshin Kim received the B.S. and M.S degree in Electrical engineering from Yonsei University, Korea in 1984 and 1986 respectively, and the Ph.D. degree in Electrical engineering from Georgia Institute of Technology, USA in 1996. He is currently a professor in

Department of Electrical Engineering, Pusan National University. His research interests include fuzzy logic controls, neuro fuzzy systems, robotics, signal analysis, and intelligent systems.

E-mail : sskim@pusan.ac.kr



Angga Rusdinar received his B. Eng. degree in Electrical Engineering from Sepuluh Nopember Institute of Technology, Indonesia, in 2001 and M. Eng. degree from the School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia in 2006. He is currently a Ph.D. program student in the

School of Electrical Engineering, Pusan National University, Korea. His research interests include robotics, robot vision, localization and navigation systems.

E-mail : angga@pusan.ac.kr