Fuzzy Logic Path Planner and Motion Controller by Evolutionary Programming for Mobile Robots

Byung-Cheol Min, Moon-Su Lee, and Donghan Kim

Abstract

A fuzzy logic controller (FLC) for mobile robots is designed in hierarchical structure. The designed FLC consists of two levels: the planner level and the motion control level. The planner level generates a path to the destination by avoiding obstacles. The singleton outputs of the planner are obtained by using lines and arc methods. The lower motion control level calculates the robot's wheel velocity so as to follow the path generated by the planner as to the current robot posture. The fuzzy singleton outputs are obtained by heuristics and tuned by evolutionary programming. The applicability of the controller is demonstrated by using robot soccer system.

Keywords: Soccer Robot, Navigation Method, Fuzzy Logic Controller (FLC,) Evolutionary Program

1. Introduction

Making a robot to move to a specific position in a constantly changing environment with internal and external constraints is the greatest goal in a navigation system. To do so, the robot should recognize the environment around itself by using the navigation plan which can make the robot to accomplish the job using a set of sensors in the varying environment [1].

The robot soccer is one of the typical examples of multi-agent system where the environments change dynamically. Robots have to cooperate with each other to compete efficiently. Also, to a certain extent, the effectiveness of the control strategies can be proved by the scores of the game. The main problems of robot soccer can be divided into three parts: exact positioning through a vision system, navigation planning using the coordinate information, and following the planned path. Among these, navigation plan represents the team's winning prospects. From the view point of navigation, robotic soccer faces a dynamic navigation problem with moving obstacles (opponent robots) and a moving target (ball). Until now, many navigation methods have been demonstrated for robotic soccer. Sim et. al. [2] proposed a four-mode control structure which has different controllers for four divided areas around a ball. Kim et. al. [3,4] proposed the uni-vector field method. Selecting two points, one in front and the other at the back of the ball, the vector angle from these two points to the current position of a robot is calculated. Then the uni-vector field is constructed by multiplying the vector angle by a certain integer. Wong et. al. [5] proposed an optimized fuzzy controller using a genetic algorithm.

Fuzzy logic is a widely used tool in mobile robotics. The path following and local obstacle avoidance behaviors are the potential fields for the application of fuzzy logic [6-7]. In most of the designs, the desired path is generated by a higher level planner. In this paper, we apply fuzzy logic to this planner level as well as to the motion control level in designing the shooting action controller of soccer robot (Figure 1).

Shooting action is the fundamental function that can be performed by each robot. It can be viewed as a posture control problem in mobile robotics' point of view. The non-holonomic constraints of wheeled mobile robots make difficult to derive stable trajectory control laws [8]. Moreover, there are several limitations because of the geometric relation among the positions of the ball, robot, and each goal area. In this paper, the followings are considered:

- The robot should approach the ball from the opposite side of the line connecting the ball and the opponent goal.
- The robot should avoid obstacles that are not very close to the ball on the field.

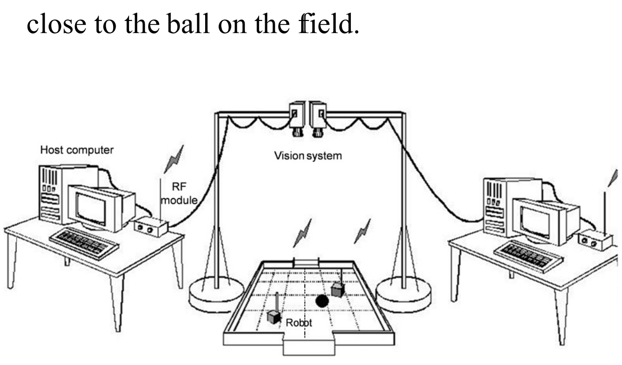


Figure 1. Overall view of the robot soccer system [9]

Corresponding Author: Donghan Kim is with the Department of Electrical Engineering, Kyung Hee University, Yongin, Korea
E-mail: donghani@khu.ac.kr
Manuscript received 20 April. 2009; revised 1 Aug. 2009; accepted 17 Sep. 2009.

Such methods as uni-vector field method, geometric calculation method, and heuristics method have been utilized to solve these constraints so far in the robot soccer competitions [2,10-13]. In this paper, we propose a fuzzy logic scheme to solve the same problem. The FLC consists of two levels: the planner and the motion controller level. Although there is the similar controller that was suggested in [14], we have approached a different way to reduce arrival time to the ball. Firstly, the fuzzy logic path planning is designed by modeling arcs and straight lines using fuzzy logic's nonlinear model mapping characteristics in this paper. However, because some path errors can be occurred when geometrical path planning does not consider kinetic characteristic, it is quite challenging to design accurate path controller. In order to solve such problems, fuzzy logic that has robust characteristic of motion controller is used. In addition, the singleton outputs of motion controller level, which are obtained in a heuristic or empirical manner, are optimized through evolutionary programming.

In Section 2, the overall fuzzy logic posture controller structure that includes a fuzzy logic path planner and a fuzzy motion controller is described. They are explained in Sections 3 and 4, respectively. In Section 5, simulations and experimental results are demonstrated. We conclude in Section 6 with some remarks on the results and future research.

2. Overall Fuzzy Control Structure

A. *Target System: Robot Soccer System*

The system (Figure 1) consists of three parts. The first one is the vision system that locates objects on the field by the global camera that is fixed above the field. Second component is the host computer which calculates strategies and decides actions for each robot. The last one is robots that follow actions given by the host computer through radio frequency (RF) communication. Since the action is represented as a number of sequential motions, sometimes host computer sends each robot's left and right wheel velocities rather than higher linguistic commands. In this paper, the fuzzy logic controller implemented in the host computer generates and sends the velocities to the robot through RF communication to achieve shooting behavior of the robot.

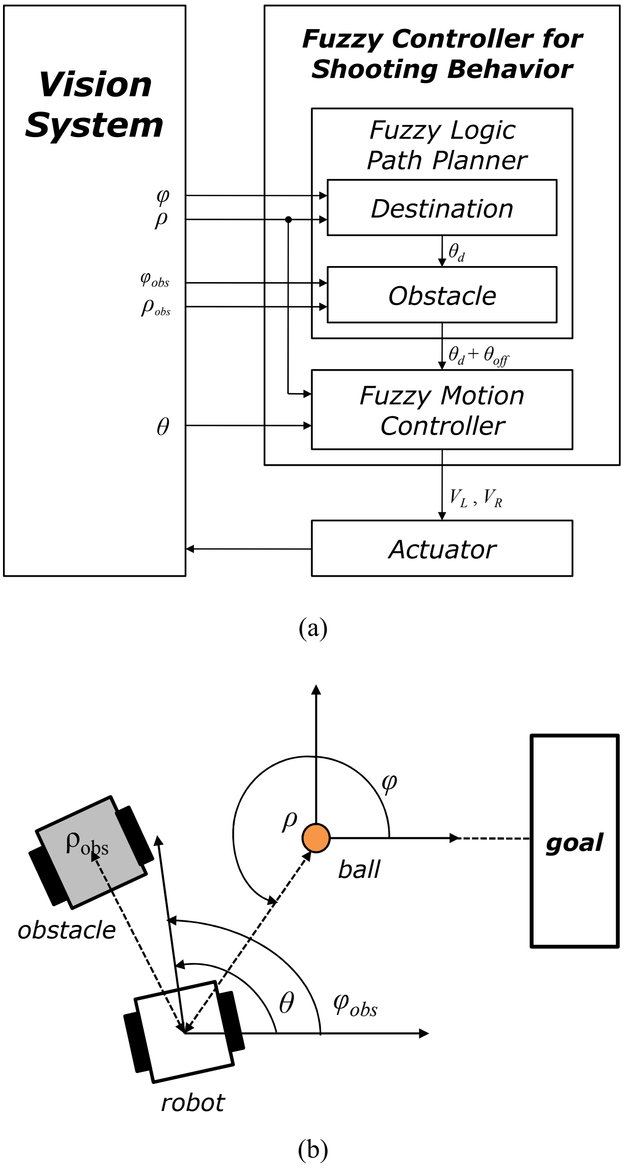
B. *Modeling of a mobile robot*

Differential-drive mobile robots with characteristics of non-slipping and pure rolling are considered. The velocity vector $Q = [v\ w]^T$ consists of the translational velocity of the center of the robot, v, and the rotational velocity, w, defined with respect to the center of the robot. The velocity vector Q and a posture vector $P = [x\ y\ \theta]^T$ are associated with the robot kinematics as follows:

$$\dot{P} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = J(\theta)Q \tag{1}$$

$$Q = [v\ \ w]^T = \begin{bmatrix} \dfrac{V_R + V_L}{2} & \dfrac{V_R - V_L}{2} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{2} & \dfrac{1}{2} \\ -\dfrac{1}{2} & \dfrac{1}{2} \end{bmatrix} \begin{bmatrix} V_L \\ V_R \end{bmatrix} \tag{2}$$

where V_L is the left wheel velocity and V_R is the right wheel velocity. The robot should be controlled to move to any posture by V_L and V_R. Hence, the fuzzy controller gives the robot a desired direction, θ_d, at the current position (x, y).



(a)

(b)

Figure 2. (a) Overall fuzzy logic path planner and motion controller structure (b) schematic diagram

C. *Overall Fuzzy Controller*

Because of the constraints mentioned in Section 1, the three posture variables (ρ,φ,θ) (Figure 2(b)) are required to achieve the shooting action. If all the variables are to be elaborated into one fuzzy rule table, the number of rules could be too large because they will increase by the factor of the number of term sets in each variable. So the controller is decomposed into two sub-controllers (Figure 2) in hierarchical manner that each takes only two variables as inputs. This will significantly reduce the number of rules and make easier to design the controller. In addition, the fuzzy motion controller and the fuzzy logic path planner are included in sub-controllers. This is for generating a global path connecting the present robot position to the ball; however, it can face the constraints. The fuzzy motion controller then commands robot wheel velocities to follow this desired path given the current robot posture.

3. Fuzzy Logic Path Planner

Fuzzy logic path planner is for generating a path globally that faces the constraints of calculating desired robot's heading angle θ_d at each relative position (ρ,φ) (Figure 2). It is again divided into two sub-blocks: the destination block that generates a path to lead to the destination (the ball) to face the first constraint in Section 1 and the obstacle block that compensate θ_d for obstacle avoidance to meet the second constraint.

A. *Destination Block*

This is for obtaining desired heading angle at each robot position. The robot's relative position to the ball is represented in polar coordinates (ρ,φ). Since the lower half plane is symmetric to X-axis, only the upper-half plane is considered.

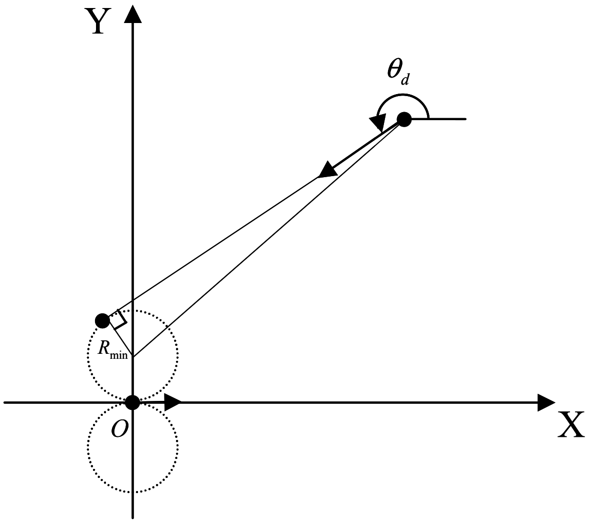


Figure 3. The desired path using lines and arcs

Figure 3 shows the basic idea of constructing the planner. The desired path is represented by lines and arcs, and the desired heading angle is defined as θ_d for the given robot position. θ_d depicted in Figure 3 can be obtained by:

$$\theta_d = -\pi + \alpha - \beta \tag{3}$$

$$= -\pi + \tan^{-1}\left(\frac{y_c - R_{\min}}{x_c}\right) - \tan^{-1}\left(\frac{R_{min}}{\sqrt{{x_c}^2 + (y_c - R_{min})^2 - R^2{}_{\min}}}\right)$$

where x_c and y_c are the positions of the current robot, and $R_{\min}$ is the turning radius, which is set to 5 cm considering the size of the ball and the robot.

With the Eq. (3), a fuzzy model is formed. In the fuzzy model, ρ and φ are assigned to the inputs based on the position of the robot in polar coordinates. Also, the output, θ_d, has singleton values obtained at the sampled positions shown in Figure 5. As a result, the input, output, and rules of the destination block are defined as follows:

1. Input space (ρ,φ) : relative position of the robot to the ball,

$$\rho \in [0\text{cm}, 60\text{cm}]$$
$$\varphi \in [0, 180 \text{ deg.}]$$

In accordance with input spaces, the input variable membership functions are depicted in Figure 4.

2. Output (θ_d): desired heading angle,

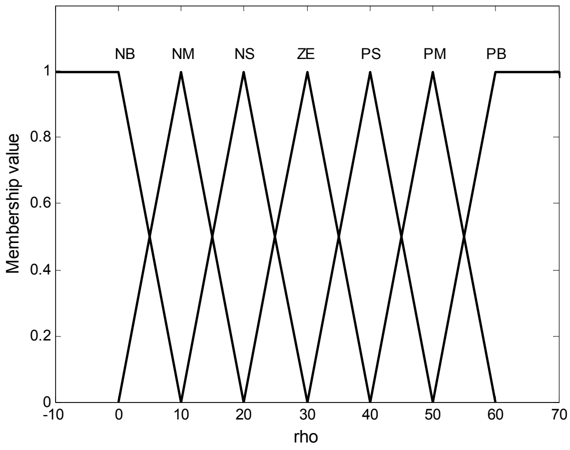
$$\theta_d \in [-180 \text{ deg.}, 180 \text{ deg.}]$$

3. Rules for destination block

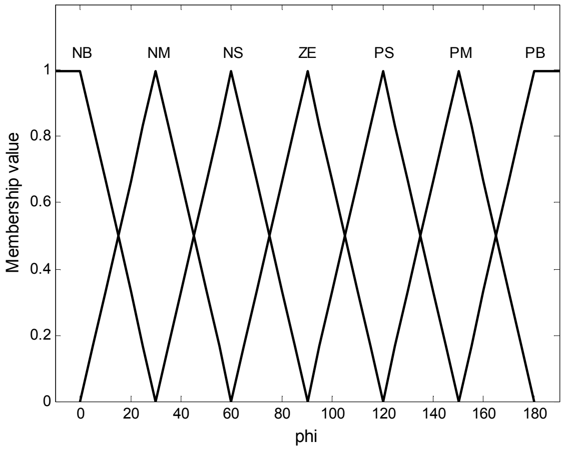
49 rules are obtained using θ_d at sampled positions as shown in Figure 5. Since input spaces are uniformly divided, the rules are sampled at the center of each input region. The resultant rule table for the destination block is in Table 1.

Table 1. Rules for destination

θ_d	ρ						
φ	NB	NM	NS	ZE	PS	PM	PB
NB	90.0	143.1	157.4	163.7	167.3	169.6	171.2
NM	120.0	158.5	-180.0	-171.0	-166.1	-163.0	-161.0
NS	160.0	172.1	-155.5	-143.6	-137.6	-134.0	-131.6
ZE	-170.0	-180.0	-126.9	-120.6	-114.3	-110.7	-108.4
PS	-140.0	-135.0	-80.0	-76.9	-71.8	-69.0	-67.4
PM	-20.0	-30.0	-34.2	-35.9	-35.5	-40.2	-45.4
PB	0	0	0	0	0	0	0



(a) ρ



(b) φ

Figure 4. Membership functions of ρ and φ according to the relative position of the robot to the ball

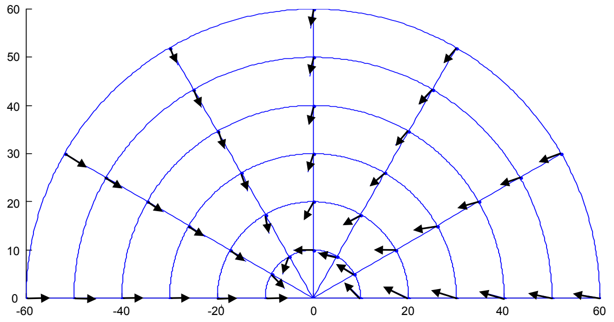


Figure 5. θ_d, desired heading angle, sampled at each region

B. *Obstacle Block*

This block is to obtain offset angle θ_{off} depicted in Figure 6, if there are any obstacles nearby.

To obtain θ_{off}, four variables such as relative velocity (V_r), relative direction (D_r), distance (d_r), and relative position (P_r, positive if obstacle in front, negative o.w.) are utilized to get θ_{esc} in the presence of obstacles.

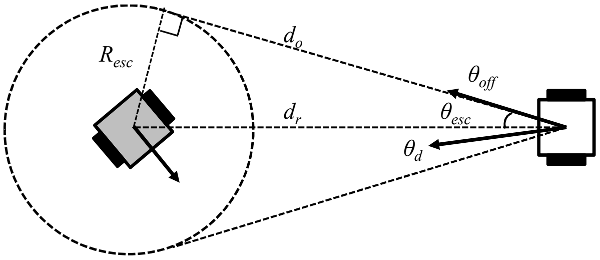


Figure 6. Obstacle avoidance scheme

θ_{esc} can be calculated with following the equation:

$$\theta_{esc} = \tan^{-1}\frac{R_{esc}}{d_o} = f(R_{esc}) \tag{4}$$

Also, those relative quantities are necessary for obtaining the escape radius (R_{esc}) to avoid stationary or moving obstacles.

However, since there are four factors we should consider for obtaining R_{esc}, it is difficult to form the FLC by using all of those factors. For this reason, those factors are divided into two FLCs and constructed in hierarchical structure shown as Figure 7.

In Figure 7, V_r and D_r are needed to obtain R_{esc}, while P_r and d_r are used to obtain the proportional gain, W_{sgn}. For example, if the obstacle is located far from the robot, W_{sgn} gets smaller and becomes 0. In contrast, if the obstacle is located nearby the robot, W_{sgn} gets bigger and reaches 2. Consequently, W_{sgn} is multiplied with θ_{esc} to produce θ_{off}.

$$\theta_{off} = W_{sgn} \times \theta_{esc}$$

where, $0 \le W_{sgn} \le 2$. As a result, the input, output, and rules for the obstacle block are defined as follows:

1. Input space (V_r, D_r, d_r, P_r): relative velocity and position of the obstacle to the robot,

$$V_r \in [-0.5, 1.5]$$
$$D_r \in [0 \text{ deg.}, 180 \text{ deg.}]$$
$$d_r \in [0\text{cm}, 90\text{cm}]$$
$$P_r \in [-0.5, 1.5]$$

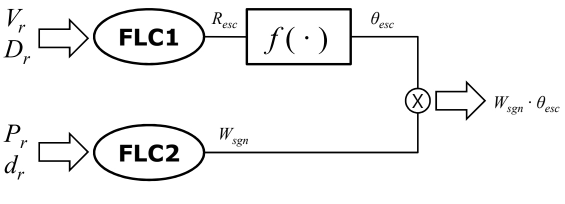
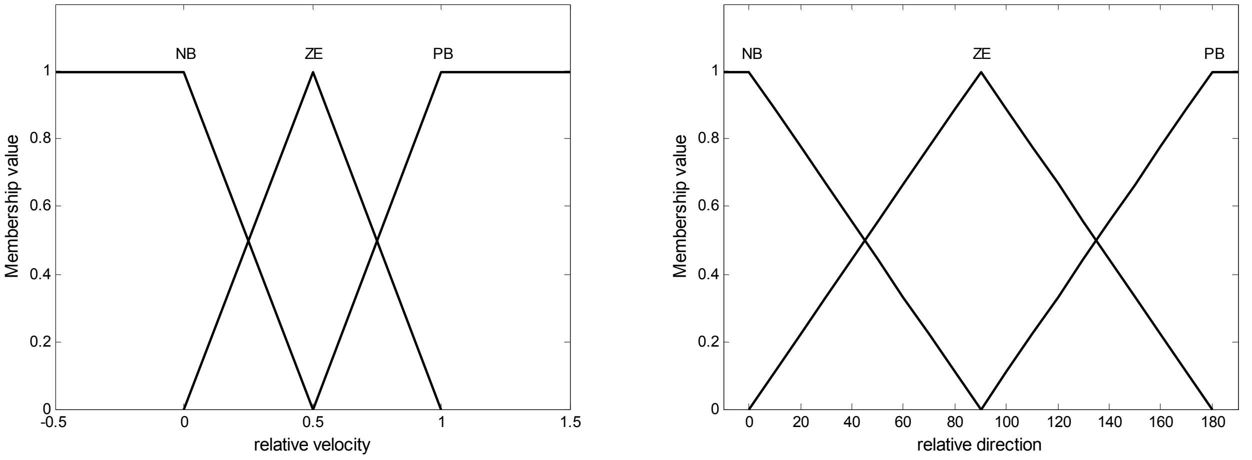
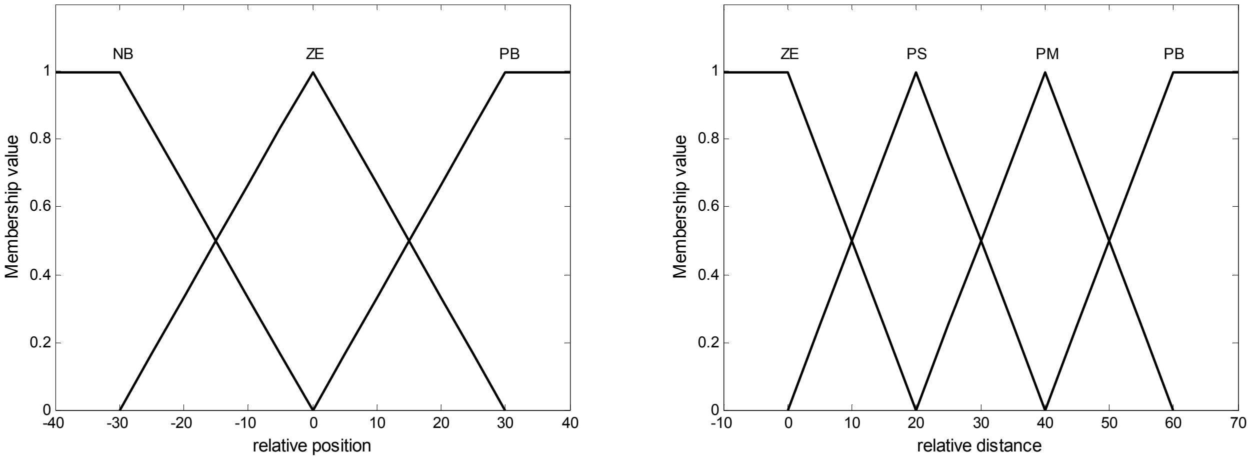


Figure 7. FLC for obstacle avoidance block



(a) V_r (b) D_r



(c) P_r (d) d_r

Figure 8. Membership functions of V_r, D_r, P_r and d_r for the obstacle block in fuzzy planner

In accordance with the input space, the input variable membership functions are depicted in Figure 8.

2. Ouput space (θ_{off}): offset angle added to θ_d,

$$\theta_{off} \in [-18O\text{deg.}, 180\text{ deg.}]$$

The resultant rule table for the obstacle block divided into FLC1 for R_{esc} and FLC2 for W_{sgn} is in Table 2.

Table 2. Rule for obstacle block, FLC1(left) and FLC2(right)

R_{esc}	D_r		
V_r	NB	ZE	PB
NB	20	20	20
ZE	20	25	30
PB	20	35	40

W_{sgn}	d_r			
P_r	ZE	PS	PM	PB
NE	0.8	0.7	0.6	0.0
ZE	1.0	1.0	0.9	0.0
PO	1.0	1.0	1.0	0.0

4. Fuzzy Motion Controller

A. *Fuzzy Motion Controller Block*

In the overall structure of Figure 2, the fuzzy motion controller block receives θ_d from the fuzzy logic path planner block and part of robot posture information (ρ, θ) through its vision. Then the motion controller block generates appropriate left and right wheel velocities to make θ follow θ_d at non-zero linear speed before ρ diminishes. So the motion controller is concerned only for heading angle θ to follow θ_d with at positive linear velocity. For this conventional problem of mobile robots, following heuristics are incorporated:

If ρ large →large
If $|\theta_e| = |(\theta_d + \theta_{off}) - \theta|$ large → $|V_L - V_R|$ large

The input, output, and rules for the fuzzy motion controller block are defined as follows:

1. Input space (ρ,θ_e): posture error of the robot to the ball and the path,

$$\rho \in [0\text{cm}, 60\text{cm}]$$
$$\theta_e \in [-120 \text{ deg.}, 120 \text{ deg.}]$$

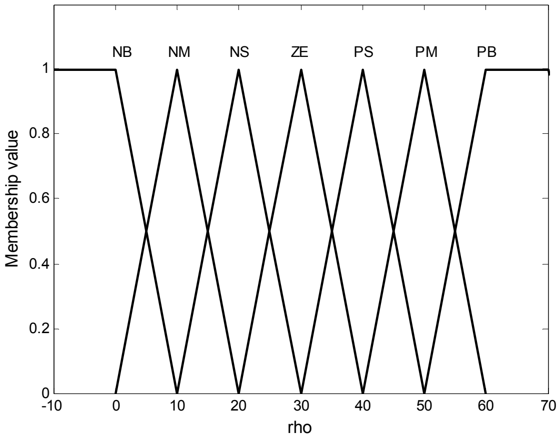
Depending on the input space, the input variable membership functions are depicted in Figure 9.

2. Output space (V_L, V_R): desired left and right wheel velocities,

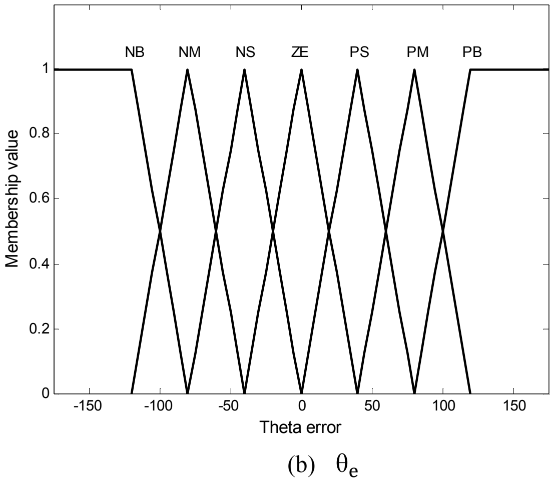
$$V_L, V_R \in [-54\text{cm/s}, 153\text{cm/s}]$$

3. Rules for fuzzy motion controller block

According to the above heuristics, 49 rules are acquired for left and right wheel velocities. Table 3 is the rule table for right wheel speed. Left wheel speed is symmetric with respect to $\varphi = 0$. In the table, one unit corresponds to 1.534cm/sec.



(a) ρ



(b) θ_e

Figure 9. Membership functions of ρ and θ_e

Table 3. Rules for right wheel

V_R	ρ						
θ_e	NB	NM	NS	ZE	PS	PM	PB
NB	-35	-27	-27	-3	-3	-3	-3
NM	-25	8	8	18	31	31	42
NS	15	15	22	35	57	67	67
ZE	30	30	50	60	90	100	100
PS	15	40	44	65	82	92	92
PM	25	51	51	61	68	68	77
PB	35	63	63	67	67	67	67

B. *Fuzzy Motion Control Block Tuning based on Evolutionary Programming*

In Section 4A, the singleton values of the motion controller block were determined with fuzzy control, which has various strong points: intuitive and simple [15]. However, most of the times, using professionals' knowledge for designing might not be the most suitable system because there will be no professionals' knowledge for newly introduced subject. In order to make up for the weak points, hybrid systems such as fuzzy system, evolutionary algorithm, and neural networks, are studied in depth.

Fuzzy system and feed-forward control of neural network have similar basic structures. The difference between those two systems is in each node's number of connections and the function. For instance, fuzzy system uses knowledge related with a plant to decide a network structure by connecting related variables and maintaining inside structure. On the other hand, neural network has a learning ability. In order to use such learning method in fuzzy system as gradient descent, which is back propagation, fuzzy system should be expressed mathematically and each of the nodes should be able to differentiate. Therefore, it is impossible to optimize the fuzzy system by using this neural network.

Evolutionary algorithm is suitable to use for optimizing a not differentiable system or a system with local solution. Because of such merits, numerous researchers applied fuzzy system in automation of system design.

For this reason, in this paper, evolutionary fuzzy system, real variables optimization method based on evolutionary algorithm, is used for tuning of fuzzy system in the path motion controller. The condition for optimization is the minimum time to reach to the target point. As shown in Figure 10, the optimization for fuzzy system uses membership function's center point and width. In order not to change the order of the membership functions, following restrictions are considered in evolutionary algorithm process.

$$A_{i-1} < A_i \qquad (i-1 < i) \tag{5}$$

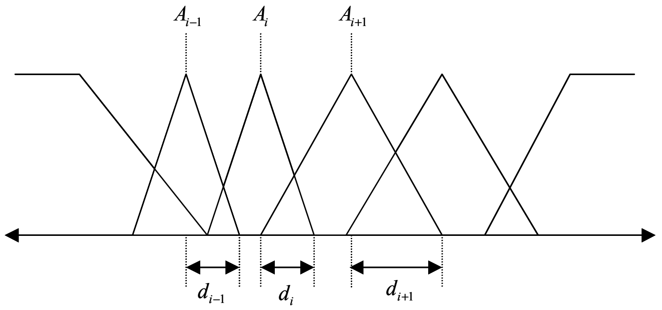


Figure 10. The optimization for fuzzy system using the triangular-shaped membership function's center point and width

Figure 11 is the flow chart that shows the learning scheme of the evolutionary program. An important feature of EPs is that the range of mutations, the *stepsize*, is not fixed but inherited. Mutation creates a new offspring x'_i from each singleton value, x_i by adding it to a Gaussian number with mean 0 and standard deviation σ_i.

$$x'_i = x_i + N_i(0, \sigma_i) \tag{6}$$

where σ_i is x_i's maximum boundary within which the designer allows to change the singleton value. By using probability selection method for selecting offspring, σ_i is set to one tenth of x_i's original value which was obtained heuristically, and total population N is set to 20. To test the tracking performance, 36 test data were used. They are 36 different starting postures. The time consumed for the robot to arrive from each point to the origin (destination) is all summed up and compared for the selection. The initial time using heuristic singleton values was 36.6 seconds. After 10,000 generations, the tuned controller reduced the time from 4.3 seconds to 32.3 seconds. Figure 12 is the results of optimizing the minimum time required to the target point while satisfying the restriction condition of Eq. (6).

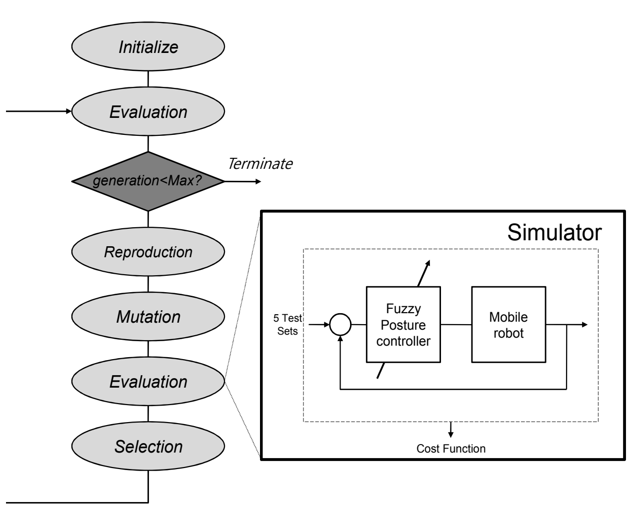
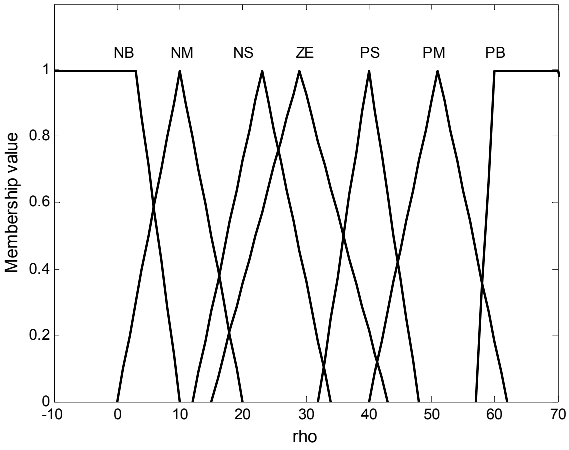
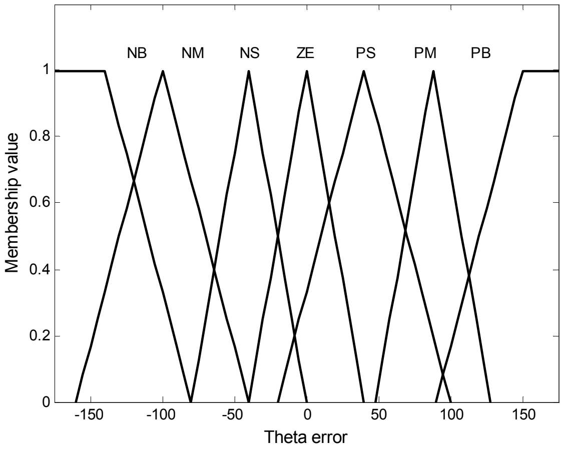


Figure 11. The flow chart of the proposed Optimization method using EP



ρ(a) ρ



(b) θ_e

Figure 12. Optimized membership functions of ρ and θ_e

5. Simulation and Experiment

A. *Simulation*

Simulation is performed based on the following robot kinematics:

$$\theta(k+1) = \theta(k) + \frac{V_R(k) - V_L(k)}{D} \cdot T_S \tag{7}$$

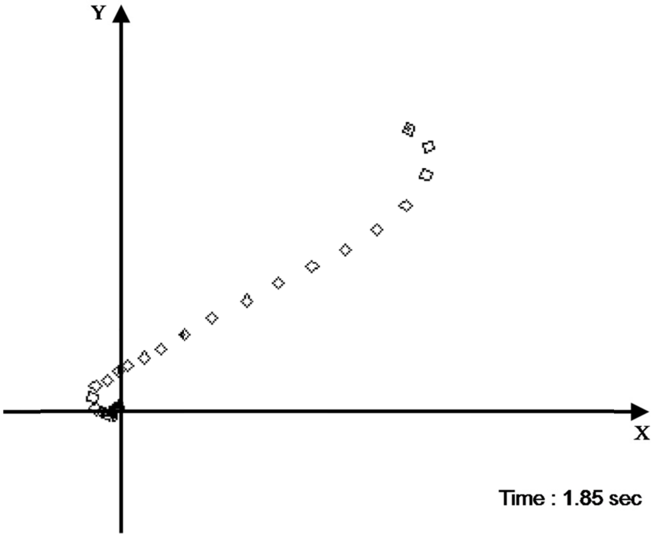
$$x(k+1) = x(k) + \frac{V_R(k) - V_L(k)}{2} \cdot \cos\frac{\theta(k) + \theta(k+1)}{2} \cdot T_S \tag{8}$$

$$y(k+1) = y(k) + \frac{V_R(k) - V_L(k)}{2} \cdot \sin\frac{\theta(k) + \theta(k+1)}{2} \cdot T_S \tag{9}$$

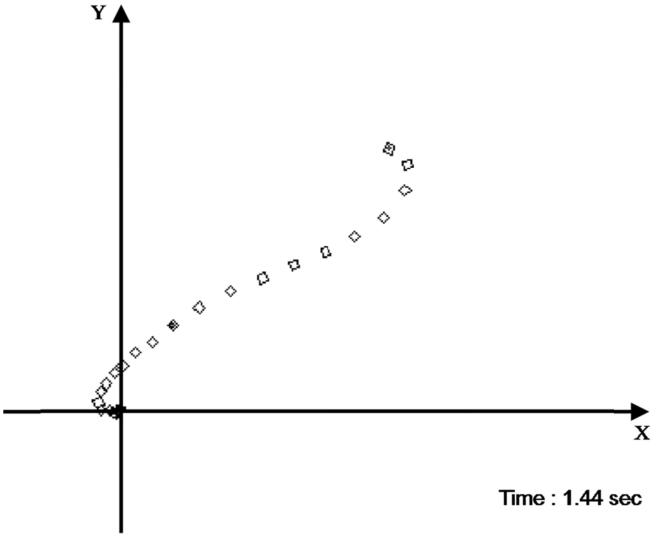
Robot's physical quantities are:

- wheelbase (D) = 6.7 cm
- wheel radius (r) = 2.0 cm
- sampling interval (T_s) = 0.5 ms

Figure 13 shows the simulation results. The tuned motion controller shows better performance by taking the linear segment part earlier to arrive at the origin in 1.44 seconds. The controller before tuned finished in 1.85 seconds. This is expectable since in applying EP, arrival time was chosen as the fitness function.



(a) Before tuning by evolutionary programming



(b) After tuning by evolutionary programming

Figure 13. Comparison of before and after tuning by evolutionary programming in simulation

B. *Experiment*

Experiments were performed using the micro robot soccer system (Soty IV, Figure 14). The system specifications include a vision board (Meteor) of 60 Hz with 640×480 resolutions and the robot, equipped with a RF module and battery, with 1.5 m/s maximum speed (Table 4). Once the vision detects the ball and robots' positions, the FLC is inferred to output the desired left and right wheel velocities. Then, those commands are sent to the robot through the RF communication module.

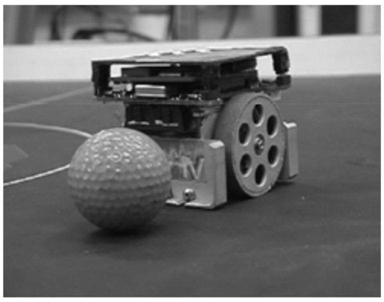


Figure 14. Soty IV soccer robot

Table 4. Specification of robot

Size	7.5cm × 7.5cm × 6.0cm
Weight	500 g
CPU	INTEL 80296SA 50MHz 32KB ROM, 256KB RAM
Motors	DC motor 8:1 Gear ratio 180 pulse/rev resolution
Power	8.4 V
RF	418/433 MHz
Comm. rate	19200bps

These processes are computed on the host computer. The robot's on-board CPU controls the speed of wheels transmitted from the host computer.

Figure 15, 16, and 17 show real experimental results of the proposed method. In Figure 15, the robot's initial values were at four different points and its direction. The robot would have certain arbitrary velocity at its target point, the origin, with facing 0 degrees.

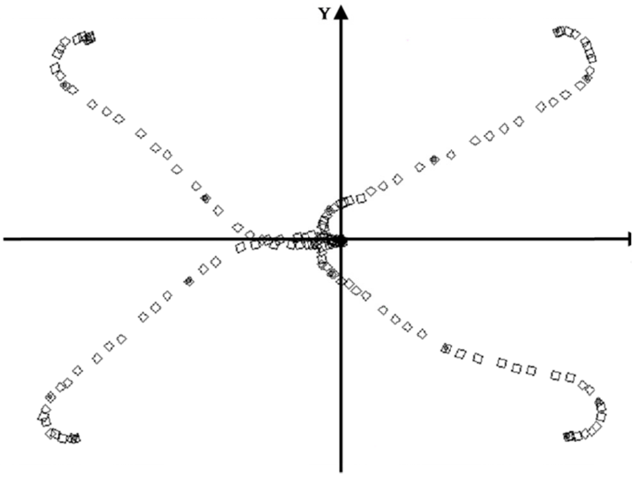
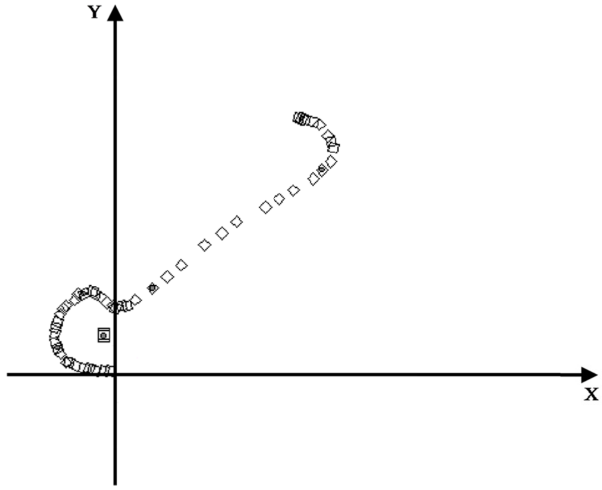


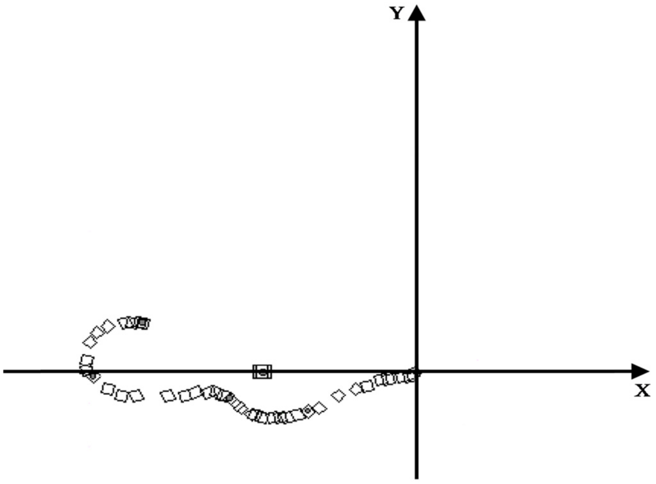
Figure 15. The Experimental result with no obstacles: four different initial points

While reaching the target point, robot's maximum velocity was 1.2m/sec robot and its velocity was 24cm/sec with 5cm rotation radius at that point. This result shows that the robot in four kinds of arbitrary initial points could reach the target point with the proposed posture control.

Figure 16 shows results of the experiment when an obstacle exists. First, several assumptions were made. The obstacle was in the robot's moving path, and the obstacle's exact size was already known. Also the obstacle was considered as a cylindrical shape. As a result, the robot detected the obstacle on its moving path, and the robot could reach the destination by avoiding the static obstacle shown in the Figure 16 (a), (b).



(a)



(b)

Figure 16. Experimental results with a static obstacle: (a) the robot starts at (100,120,-10°) (b) the robot starts at (-110,20,180°)

The next experiment demonstrates the effectiveness of the proposed fuzzy logic path planner and motion controller in the situation in which moving obstacles exist. At this situation, the obstacle's cylindrical shape and its velocity, 24cm/sec, is given. In Figure 17, the trajectory of the moving obstacle is depicted as a dotted line, while the trajectory of the robot is depicted as a solid line. As those trajectories show, the robot could avoid the moving obstacle that is coming toward the robot's moving path and reach the destination even though it is difficult for the robot to avoid the moving obstacle.

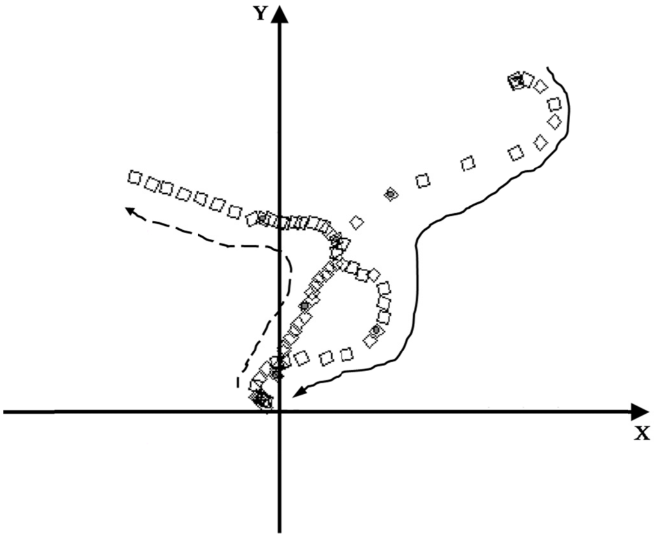


Figure 17. Experimental result: when the obstacle is moving towards the robot's path

6. Conclusions

A fuzzy logic controller (FLC) for mobile robot path control was suggested and applied to shooting action of soccer robot. Shooting action was tackled as a posture control problem with a constraint that the robot should approach the ball in such a way that the ball goes to the opponent team area. It should avoid nearby obstacles as well. Two hierarchical sub-blocks of the FLC structure reduced the number of rules. The planner was synthesized to generate paths that consist of line and arc. The singleton output for the planner could be obtained accordingly. The heuristically obtained motion controller's singleton output values were tuned with evolutionary programming. Simulation and experimental results showed the applicability of the proposed control scheme. Fine tuning and optimization for the obstacle avoidance block is needed in future work. The motion of the ball should be also considered for optimization.

7. Acknowledgment

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (MEST) (R01-2008-000-20352-0).

7. References

[1] Ching-Chang Wong, Chi-Tai Cheng, Kai-Hsiang Huang, and Yu-Ting Yang, "Fuzzy Control of Humanoid Robot for Obstacle Avoidance," *International Journal of Fuzzy Systems*, Vol. 10, No. 1, March 2008.

[2] H. -S. Shim, H.-S. Kim, M.-J. Jung, I.-H. Choi, J.-H. Kim, "Designing distributed control architecture for cooperative multi-agent system and its real-time application to soccer robot," *Robotics and Autonomous Systems*, V01.21, No. 2, pp. 149-166, 1997.

[3] Y.-J. Kim, D.-H, Kim and J.-H. Kim, "Evolutionary Programming-Based Vector Field Method for Fast Mobile Robot Navigation," *Proceedings of Second Asia-Pacific Conference on Simulated Evolution and Learning*, Canberra, Dec. 1998.

[4] Y.-J. Kim, D.-H. Kim and J.-H. Kim, "Evolutionary Programming-Based Uni-Vector Field Method for Fast Mobile Robot Navigation," *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1999.

[5] Ching-Chang Wong, Hoi-Yi Wang, Shih-An Li and Chi-Tai Cheng, "Fuzzy Controller Designed by GA for Two-wheeled Mobile Robots," *International Journal of Fuzzy Systems*, Vol. 9, No. 1, March 2007.

[6] S. Ishikawa, "A method of autonomous mobile robot navigation by using fuzzy contro1," *Advanced Robotics*, Vol. 9, NO. 1, pp.29-52, 1995.

[7] J. S. Choi, J. S. Joo, "FLC for Tracking Problem of Mobile Robot with Obstacle Avoidance," *Intelligent Control Class term project report*, Dept. of Electrical Engineering, KAIST, 1996.

[8] J.-M. Yang and J.-H. Kim, "Sliding Mode Control for Trajectory Tracking of Nonholonomic Wheeled Mobile Robots," *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 3, Jun. 1999.

[9] Nishant Agrawal, Tamhant Jain," Real Time Vision Guided Reactive Strategies for Micro Robot Soccer", *B.S. thesis, Dept. Electrical Eng., Indian Institute of Technology*, Kanpur, India, 2000

[10] Y.-J. Kim, D.-H, Kim and J.-H. Kim, "Evolutionary Programming-Based Vector Field Method for Fast Mobile Robot Navigation," *Proceedings of Second Asia-Pacific Conference on Simulated Evolution and Learning*, Canberra, Dec. 1998.

[11] Y.-J. Kim, D.-H. Kim and J.-H. Kim, "Evolutionary Programming-Based Uni-Vector Field Method for Fast Mobile Robot Navigation," *Lecture Notes in Artificial Intelligence*, Springer-Verlag, 1999.

[12] G. Springer, P. Hannah, R. J. Stonier, S. Smith, P. Wolfs, "Simple strategies for collision-avoidance in robot soccer," *Robotics and Autonomous Systems*, V01.21, No. 2, pp.191-196, 1997.

[13] *Proceedings of the 1st Workshop on Soccer Robotics (Korean)*, Federation of International Robot-soccer Association (FIRA), 1999.

[14] M.-J. Jung, H. -S. Kim, H. -S. Shim and J. -H. Kim, "Fuzzy Rule Extraction for Shooting Action Controller of Soccer Robot," *1999 IEEE International Fuzzy Systems Conference Proceedings*, pp. 556-561,1999.

[15] Li-Xin Wang, J. M. Mendel, "Generating Fuzzy Rules by Learning from Examples," *IEEE Transactions on Sys., Man, and Cybernetics*, Vol. 22, No. 6, pp.1414-1427,1992.

Byung-Cheol Min received his B.S. degree in Electronics Engineering from the Kyung Hee University, Yongin, Korea, in 2008. He is currently pursuing his M.S. degree in Electronics and Radio Engineering from Kyung Hee University. His current research interests include intelligent control and navigation control.

Moon-Su Lee received his B.S. degree in Information & Control Engineering from the University of Kwang Woon, Seoul, Korea in 1998 and M.S. degree in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Deajon, Korea in 2000. Since then, he has been with telematics research division, ETRI, Korea, as a research senior member.

Dong-Han Kim received his B.S., M.S and Ph. D. degrees in Electrical Engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 1995, 1998 and 2003, respectively. He has been with the Department of Electrical Engineering at Kyung Hee University, where he is currently an assistant professor. He was a post doctoral fellow at University of Illinois at Urbaba-Champaign in 2003. He was the recipient of the first award of a humanoid robot competition at the FIRA robot soccer in 2002. His current research interests are in the area of a multi-agent robot system.