

미지 환경에서 이동로봇의 주행 알고리즘

A Navigation Algorithm for Mobile Robots in Unknown Environments

이현재* · 최영규**

Hyun-Jae Yi, Young-Kiu Choi

* 부산대학교 대학원 전기공학과

** 부산대학교 공과대학 전자전기통신공학과

요 약

본 논문에서는 자율이동로봇이 주위 환경을 알지 못하는 상황에서 목표점까지 안전하게 주행하게 하는 주행 알고리즘에 대해 연구한다. 주행 알고리즘에서 가장 고려해야할 부분이 장애물 회피 알고리즘인데, 본 논문에서는 장애물 회피 알고리즘인 VFH(Vector Field Histogram) 알고리즘과 퍼지 알고리즘을 조합하여 여러 가지 형태의 환경에서 최적의 성능을 내도록 한다. 로봇이 처한 환경에 따라 상위 레벨의 supervisor가 위의 두 가지 장애물 회피 알고리즘 중 적절한 것을 선택하도록 조합하고, 다양한 환경에서 모의실험을 수행함으로써 제안된 로봇주행 알고리즘의 성능을 검증한다.

Abstract

This paper deals with problems of safe and efficient navigation algorithm for autonomous mobile robots in unknown environments. Since the obstacle avoidance algorithms are very important in mobile robot navigation, two obstacle avoidance algorithms: VFH(vector field histogram) algorithm and a fuzzy algorithm are combined to have optimal performance in various environments. And a upper-level supervisor is to select the proper one from VFH algorithm and the fuzzy algorithm according to the situations the robot faces. Computer simulation results show the effectiveness of the proposed navigation algorithm for autonomous mobile robots.

Key words : Mobile robots, Navigation algorithm, Unknown environments, Supervisor

1. 서 론

로봇은 다양한 분야에서 그 필요성이 대두되고 있으며, 산업용 로봇, 무인 탐사 로봇, 서비스 로봇, 청소 로봇, 휴먼 로봇 등의 다양한 형태로 이용되고 있고 최근에 차세대 사업 중 하나로 선정되기도 하였다. 과거의 로봇 연구는 단순, 반복적인 작업을 수행하는데 국한되어 진행되어 왔지만, 인공지능 알고리즘의 발달과 그것의 구현을 가능하게 해주는 기계, 전자 기술의 발달로 현재에는 로봇이 스스로 판단하고 행동하는 지능형 로봇에 대한 연구가 활발하다. 그 중 자율이동로봇(autonomous mobile robot)에 속하는 연구는 지도 생성(map generation), 위치파악(localization), 장애물 회피(obstacle avoidance)알고리즘, 지도로부터 적절한 경로를 생성해 내는 경로계획(path planning) 등으로 진행되고 있다. 특히 경로계획에 관한 연구는 많은 방법으로 진행되고 있는데, 크게 영역 전체를 검색하여 경로를 생성하는 전역경로계획(global path planning)과 근처 장애물의 변화에 따라 수시로 변화하는 지역경로계획(local path planning)으로 나눌 수 있다.

전역경로계획은 로봇이 주행해야할 환경에 대한 정보를

미리 주고 목표점까지의 최적의 경로를 설정하여 로봇을 주행하게 한다[1-3]. 이러한 전역경로계획은 이미 알고 있는 환경에서 장애물과 충돌 없는 주행을 가능하게 하지만 환경이 변화하거나 환경에 대한 정보가 잘못되면 로봇이 목표점까지 도달할 수 없는 경우가 발생한다. 이에 비해 지역경로계획은 주행해야 할 환경에 대한 정보를 가지고 있지 않은 상태에서 로봇에 장착된 센서의 입력을 통해 주어진 목표점까지 주행하게 한다. 지역경로계획을 갖춘 로봇은 다양하게 변화하는 환경에서도 높은 자율성을 가지고 주행할 수 있다. 기존의 지역 경로계획에 의한 주행 알고리즘은 주로 가상의 힘이라는 개념을 도입한 potential field, 퍼지 시스템, 신경회로망 등에 기초하고 있다[4-6]. 현재 주로 연구되고 있는 청소 로봇이나 탐사 로봇은 주위 환경에 대한 모든 정보를 가지고 있지 않다. 따라서 로봇은 주위 환경에 대한 정보를 획득하여 자율적으로 자신이 나아가야할 경로를 찾아가야만 한다.

본 논문은 자율이동로봇이 주위 환경을 알지 못하는 상황에서 목표점까지 안전하게 도달하게 하는 주행 알고리즘의 관점에서 연구하였다. 주행 알고리즘에서 가장 고려해야할 부분이 장애물 회피 알고리즘인데, 본 논문에서는 장애물 회피 알고리즘인 VFH(Vector Field Histogram)알고리즘[7]과 퍼지 알고리즘[8,9]을 조합하였다. 즉, 로봇이 처한 여러 가지 환경에서 VFH 알고리즘과 퍼지 알고리즘을 적절히 선택하는 supervisor를 상위 레벨에 두어 전체적으로 최적의 성능을 내도록 하였다. 그리고 다양한 환경에서의 모의실험을 수

접수일자 : 2006년 1월 23일

완료일자 : 2006년 4월 23일

감사의 글 : 이 논문은 부산대학교 자유과제 학술연구비(2년)에 의하여 연구되었음

행함으로써 제안된 로봇주행 알고리즘의 성능을 검증하였다.

2. 자율이동로봇 시스템

본 논문에서 로봇을 차륜형(differential wheeled type) 구동방식으로 채택하였고 로봇의 위치 좌표를 알기 위해 기구학을 이용하였다. 그리고 로봇이 주행해야 하는 작업공간의 환경지도를 바탕으로 로봇이 인식할 수 있는 영상지도로 바꾸기 위한 격자형 환경모델을 사용해서 환경지도를 2차원 격자 지도로 변환한다.

2.1 자율이동로봇의 기구학

자율이동로봇의 구동부는 차륜형이며 좌측 바퀴와 우측 바퀴가 독립적으로 제어되는 방식으로 좌측바퀴와 우측바퀴의 각속도를 각각 ω_l , ω_r 이라고 하면 non-slipping조건에 의해 좌측 바퀴와 우측 바퀴의 선속도 v_l , v_r 은 식(1)과 같다.

$$v_r = r\omega_r, v_l = r\omega_l \quad (1)$$

단, r 은 바퀴의 반지름이다.

그리고 로봇의 각속도는 두 바퀴의 차동 속도에 의해 제어된다. 로봇의 기구학 해석을 위해 그림 1과 같이 2차원 평면상에 구성된 좌표계를 설정하고 로봇은 지면과 미끄러짐 없이 이동한다고 가정한다.

그림 1에 나타난 바와 같이 로봇의 좌측 바퀴와 우측바퀴 사이의 중심점을 C 라고 하면, C 지점에서의 로봇의 선속도 v_c 및 각속도 ω_c 는 식(2)와 (3)에 의해 각각 구해진다.

$$v_c = \frac{v_r + v_l}{2} = r \frac{\omega_r + \omega_l}{2} \quad (2)$$

$$\omega_c = \frac{v_r - v_l}{L} = r \frac{\omega_r - \omega_l}{L} \quad (3)$$

단, L 은 양 바퀴 사이의 거리이다.

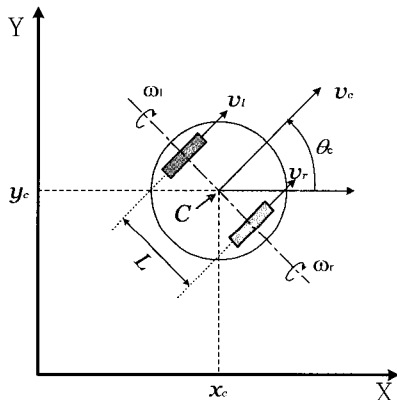


그림 1. 이동로봇의 좌표계
Fig. 1. Coordinate system for mobile robot

또한 로봇의 회전반경 R 은 식(4)에 의해 구하여진다.

$$R = \frac{L}{2} \cdot \frac{(\omega_r + \omega_l)}{(\omega_r - \omega_l)} \quad (4)$$

만약 $\omega_r = \omega_l$ 이면 회전반경 R 은 ∞ 가 되어 로봇이 직진하고 있음을 의미한다. $\omega_r = -\omega_l$ 이면 회전반경 R 은 0 이 되어 로봇이 C 지점을 중심으로 제자리에서 회전하고 있음을 나타낸다. 이와 같이 차륜형 구동방식으로 설계된 로봇의 경우 회전반경이 0 이 될 수 있으므로 좁은 환경에서도 주행이 가능하다는 장점을 지니고 있다.

그림 1과 같이 설정된 전역 좌표계에서 자율이동로봇의 상태는 로봇의 위치와 방향각으로 구성된 자세(posture) 벡터 $P = [x_c \ y_c \ \theta_c]^T$ 로 나타내진다. 로봇에서 바퀴의 회전 속도와 그에 따른 자세의 관계를 나타낸 식을 기구학 식이라 부르며, $[\dot{x}_c \ \dot{y}_c \ \dot{\theta}_c]^T$ 와 $[v_c \ \omega_c]^T$ 는 기구학 식(5)에 의해 연관 지워진다.

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{\theta}_c \end{pmatrix} = \begin{pmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_c \\ \omega_c \end{pmatrix} \quad (5)$$

단, (x_c, y_c) 는 전역 좌표계에서 로봇의 위치에 해당하는 X, Y 축 좌표값이며, θ_c 는 X 축을 기준으로 한 로봇의 방향각을 나타낸다[10].

설정된 좌표계에 대한 기구학 방정식으로부터 로봇의 자세를 식(6)~(11)과 같이 구할 수 있다.

1) $\omega_c = 0$ 일 경우

$$x_c^{k+1} = x_c^k + v_c t_s \cos(\theta_c^k) \quad (6)$$

$$y_c^{k+1} = y_c^k + v_c t_s \sin(\theta_c^k) \quad (7)$$

$$\theta_c^{k+1} = \theta_c^k \quad (8)$$

2) $\omega_c \neq 0$ 일 경우

$$x_c^{k+1} = x_c^k + \frac{v_c}{\omega_c} [\sin(\theta_c^{k+1}) - \sin(\theta_c^k)] \quad (9)$$

$$y_c^{k+1} = y_c^k - \frac{v_c}{\omega_c} [\cos(\theta_c^{k+1}) - \cos(\theta_c^k)] \quad (10)$$

$$\theta_c^{k+1} = \theta_c^k + \omega_c t_s \quad (11)$$

단, t_s 는 샘플링 시간이다.

현재의 로봇 위치를 알기 위해서 모터와 직결된 엔코더의 위치 신호를 받아 식 (6)~(11)에 의해 로봇의 위치를 계산하는데 이를 dead-reckoning 방법이라고 한다.

2.2 전체 주행 알고리즘

자율이동로봇이 정해진 목표점까지 이동하는데 있어 고려해야 될 사항은 출발점, 목표점, 로봇의 실시간적인 위치 및 장애물의 위치이다. 그리고 지역경로계획에서 자율이동로봇은 목표점까지 주행하는 동안 장애물을 회피함과 동시에 목표점을 향하여 이동하여야 한다.

본 연구에서는 장애물 회피 행위(obstacle avoider behavior)와 목표점 도달행위(goal seeker behavior)를 퍼지 알고리즘과 VFH 알고리즘을 통해 구현하였고, 주위 환경 즉 장애물의 형태에 따라 더 나은 성능을 보이는 알고리즘을 선택하는 supervisor를 두어 목표점까지의 도달 시간을 최소화 하였다. 장애물을 인식하기 위해 초음파 센서를 사용하였으

며, 전체 주행 알고리즘에 대한 순서도는 그림 2와 같다.

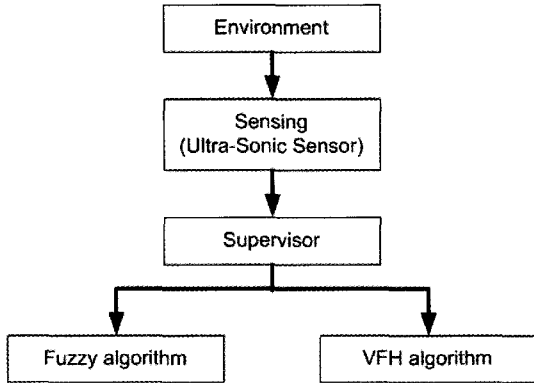


그림 2. 전체 주행 알고리즘
Fig. 2. Overall navigation algorithm

3. 퍼지 알고리즘

자율이동로봇의 주행을 위한 퍼지 알고리즘의 실제적 설계 방법과 자율이동로봇으로의 적용에 대해 설명한다. 자율이동로봇의 행위는 크게 목표점을 찾아가는 목표점 도달 행위와 장애물을 회피하는 장애물 회피 행위로 나눌 수 있다. 다음은 각각의 행위에 대한 내용이다.

3.1 목표점 도달 행위

목표점 도달 행위는 로봇이 주행할 때 목표점이 로봇의 진행 방향에서 왼쪽에 존재하면 로봇은 왼쪽으로 이동하게 하고 오른쪽에 존재하면 오른쪽 방향으로 이동하게 한다. 또한 로봇이 목표점과 멀리 떨어져 있으면 빠르게 움직이도록 하며 목표점에 가까이 접근하면 로봇은 낮은 속도로 주행하게 한다.

목표점 도달 행위에서 로봇과 목표점 사이 각도의 입력 소속함수(membership function)는 그림 3과 같이 설정하였고, 로봇과 목표점 사이 거리의 입력 소속함수는 그림 4와 같이 설정하였다. 그림 3에서 LB 는 Left Big 을 나타내고, LS 는 Left Small, Z 는 Zero, RS 는 Right Small, RB 는 Right Big 을 나타낸다.

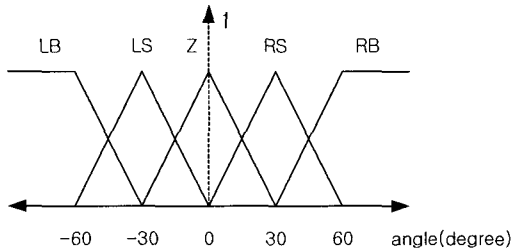


그림 3. 로봇과 목표점 사이 각도 소속함수
Fig. 3. Membership functions of angles between the robot and goal

그림 4 에서 VN 은 Very Near, SN 은 Small Near, ME 는 Medium, FA 는 Far, VF 는 Very Far이다.

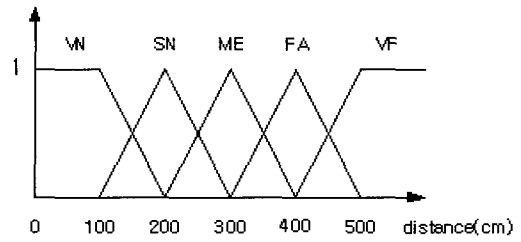


그림 4. 로봇과 목표점 사이 거리 소속함수
Fig. 4. Membership functions of distances between the robot and goal

그림 5는 로봇의 각속도 퍼지출력 소속함수를 나타내고, 그림 6은 선속도 퍼지 출력 소속함수를 나타낸다. 그림 6에서 Z 는 Zero, VS 는 Very Slow, S 는 Slow, M 은 Medium, F 는 Fast이다.

이동로봇의 중심점과 목표점이 이루는 각과 로봇의 진행 방향의 각의 차이에 따라 출력 각속도 값이 결정되며, 이 값이 양수일 경우는 이동로봇은 왼쪽으로 회전하며 음수일 경우는 오른쪽으로 회전한다.

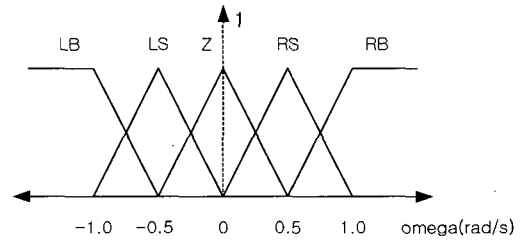


그림 5. 로봇의 각속도 소속함수
Fig. 5. Membership functions of angular velocities of the robot

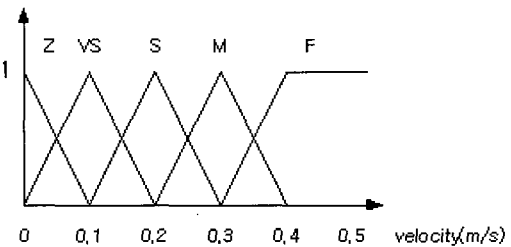


그림 6. 로봇의 선속도 소속함수
Fig. 6. Membership functions of linear velocities of the robot

목표점 도달 행위에 관한 퍼지 규칙은 표 1, 2와 같이 설정하였다.

퍼지 시스템을 이용한 목표점 도달 행위는 일반적인 로봇 주행 행위에 비해 퍼지 규칙 설정이 용이하다. 또한 로봇은 목표점으로 안전하게 주행하기 위하여 퍼지 규칙을 실시간으로 적용하며, 로봇 자신의 현재 상태를 기억한다. 입력 소속함수는 삼각형 소속 함수를 사용하였으며 추론 과정은 Mamdani의 max-min방법을 사용하였다. 로봇의 출력 값을 계산하기 위한 비퍼지화 방법은 무게중심법을 사용하였다 [11].

표 1. 목표점도달행위 각속도 퍼지 규칙
Table 1. Fuzzy rules for angular velocity in target tracking behavior

Angle \ Dist	LB	LS	Z	RS	RB
VN	LB	LB	Z	RB	RB
SN	LB	LB	Z	RB	RB
ME	LB	LB	Z	RB	RB
FA	LB	LS	Z	RS	RB
VF	LS	LS	Z	RS	RS

3.2 장애물 회피행위

이동로봇은 목표점을 향해 주행하는 동시에 안전하게 장애물을 회피하여야 한다. 이를 위한 장애물 회피 행위도 퍼지 알고리즘으로 구현하였다. 로봇은 9개의

표 2. 목표점도달행위 선속도 퍼지 규칙
Table 2. Fuzzy rules for linear velocity in target tracking behavior

Angle \ Dist	LB	LS	Z	RS	RB
VN	VS	VS	Z	VS	VS
SN	S	S	VS	S	S
ME	M	S	S	S	M
FA	F	M	M	M	F
VF	F	F	F	F	F

초음파 센서를 가지고 장애물을 감지하며, 주행 시 장애물이 로봇 진행방향을 기준으로 하여 오른쪽에 존재하면 로봇은 왼쪽으로 회피하고 왼쪽에 존재하면 반대 방향으로 회피한다.

장애물 회피행위의 입력 소속 함수는 그림 7, 8 과 같으며 로봇이 감지하는 초음파 센서는 로봇의 진행방향을 중심으로 우측으로부터 좌측으로 22.5°의 간격으로 -90° ~ 90° 사이에 9개의 센서를 가동하였다.

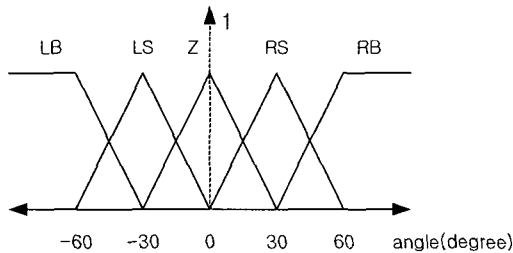


그림 7. 로봇과 장애물 사이 각도 소속함수
Fig. 7. Membership functions of angles between the robot and obstacle

초음파 센서는 최대 50cm 까지 감지하도록 하였다. 이동로봇이 주행할 때 전 단계의 로봇의 위치와 방향이 다음 단계의 로봇의 자세에 영향을 미치므로 초음파 센서의 감지 거리를 짧게 하여 매 순간마다 로봇이 처한 환경에 대하여 가장 적절한 출력을 가지게 하였다.

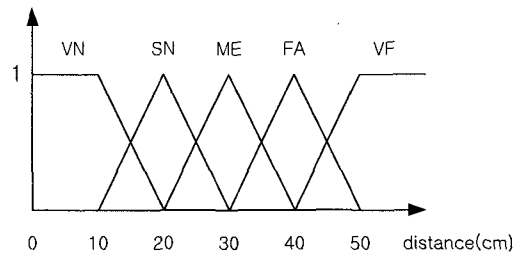


그림 8. 로봇과 장애물 사이 거리 소속함수
Fig. 8. Membership functions of distances between the robot and obstacle

그림 9, 10 은 퍼지 출력을 나타내며, 로봇이 장애물을 만날 시 최대 0.3[m/s]로 주행하도록 하였다. 로봇이 목표점과 멀리 떨어져 있을 때, 최대 0.5[m/s]로 주행하다가 장애물을 감지하면 0.3[m/s]이내로 속도를 줄여 안전하게 장애물을 회피하도록 하였다.

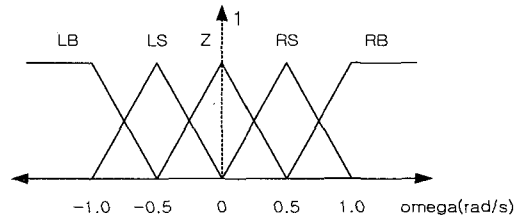


그림 9. 로봇의 각속도 소속함수
Fig. 9. Membership functions of angular velocities of the robot

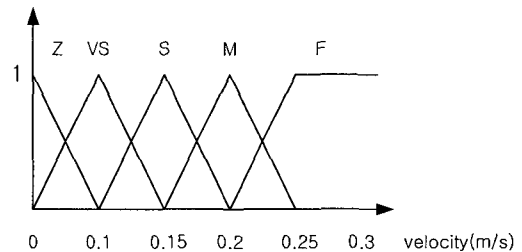


그림 10. 로봇의 선속도 소속함수
Fig. 10. Membership functions of linear velocities of the robot

장애물 회피행위에 관한 퍼지 규칙은 표 3, 4와 같이 설정되었다.

표 3. 장애물회피행위 각속도 퍼지 규칙
Table 3. Fuzzy rules for angular velocity in obstacle avoidance behavior

Angle \ Dist	LB	LS	Z	RS	RB
VN	RB	RB	RB	LB	LB
SN	RB	RB	RB	LB	LB
ME	RB	RB	RB	LB	LB
FA	RS	RS	RS	LS	LS
VF	RS	RS	RS	LS	LS

표 4. 장애물회피행위 선속도 퍼지 규칙
Table 4. Fuzzy rules for linear velocity in obstacle avoidance behavior

Angle \ Dist	LB	LS	Z	RS	RB
VN	VS	VS	VS	VS	VS
SN	S	S	S	S	S
ME	F	M	M	M	F
FA	F	M	M	M	F
VF	F	F	F	F	F

4. VFH 알고리즘

VFH(Vector Field Histogram) 알고리즘은 두 단계의 데이터 축소를 행하므로 총 3단계의 데이터 표현이 존재하게 되는데, 가장 뒤 단계는 확률 격자 지도이고 중간 단계는 극 히스토그램(polar histogram)이며 가장 낮은 단계는 방향제어 기준 값이다[7].

4.1 첫 번째 데이터 축소

첫 번째 데이터 축소 단계는 격자 지도의 활성 영역에서 극 히스토그램을 만드는 단계이다. VFH 알고리즘에서 하나의 윈도우가 로봇을 가운데에 두고 격자 지도를 움직이는데 그 영역을 활성 영역이라 한다. 활성 영역 안에 있는 cell 들은 방향 성분 $\beta_{i,j}$ 와 크기 성분 $m_{i,j}$ 를 가지는 벡터로 치환 된다.

$$\beta_{i,j} = \tan^{-1} \frac{y_j - y_0}{x_j - x_0} \quad (12)$$

$$m_{i,j} = (c_{i,j}^*)^2 (a - bd_{i,j}) \quad (13)$$

단, a, b 는 양의 상수이고, $c_{i,j}^*$ 는 cell(i, j)의 확률 값이며, $d_{i,j}$ 는 cell(i, j)와 로봇의 거리, (x_0, y_0) 는 이동로봇의 좌표, (x_i, y_j) 는 cell(i, j)의 좌표, $\beta_{i,j}$ 는 cell(i, j)에서 부터 로봇까지의 각도이다.

극 히스토그램은 그림 11 에서처럼 활성 영역을 $\beta_{i,j}$ 의 값에 따라 n개의 영역으로 나누고, 각각의 영역에 있는 $m_{i,j}$ 들을 더한 값을 극 장애물 밀도 h_k 로 나타내는 것이다.

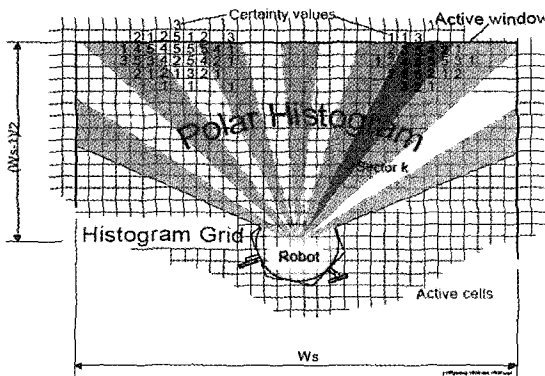


그림 11. 활성 영역에서 만들어진 극 히스토그램
Fig. 11. Mapping of active cells onto the polar histogram

$$h_k = \sum_{i,j} m_{i,j} \quad (14)$$

$$k = INT\left(\frac{\beta_{i,j}}{a}\right) \quad (15)$$

단, $a=360 / n$, n = 극 히스토그램 영역 개수

격자 지도의 불연속적인 특성 때문에 발생하는 오차를 줄이기 위해 극 히스토그램을 식(16)에서 다시 보정하여 준다.

$$h'_k = \frac{h_{k-l} + 2h_{k-l+1} + \dots + lh_k + \dots + 2h_{k+l-1} + h_{k+l}}{2l+1} \quad (16)$$

4.2 두 번째 데이터 축소

두 번째 데이터 축소는 로봇의 진행 방향을 구하는 과정이다. 그림 12는 로봇의 극 히스토그램을 보여주고 그림 13은 어떠한 threshold값을 기준으로 봉우리(peaks)와 골짜기(valleys)를 가지게 됨을 보여준다.

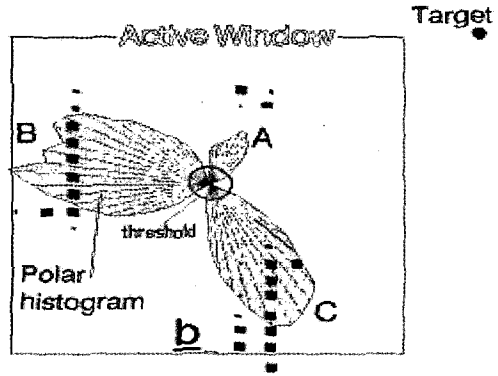


그림 12. 극 히스토그램
Fig. 12. Polar histogram

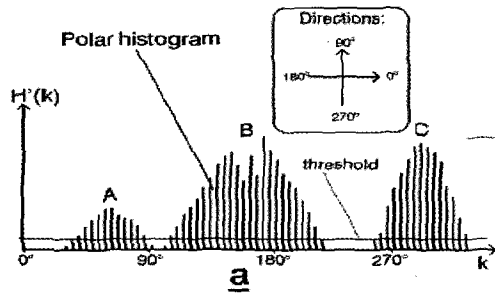


그림 13. 직각좌표계로 옮긴 극 히스토그램
Fig. 13. Polar histogram in Cartesian coordinates

이렇게 생성된 골짜기들을 후보 골짜기(candidate valleys)라 한다. 그림 14는 봉우리와 골짜기 모습을 로봇을 중심으로 보여주고 있다. sector B는 봉우리 부분이고, sector A는 골짜기 부분이다. 로봇은 구해진 sector A 구간 중 한 구간으로 진행 방향을 결정하게 된다.

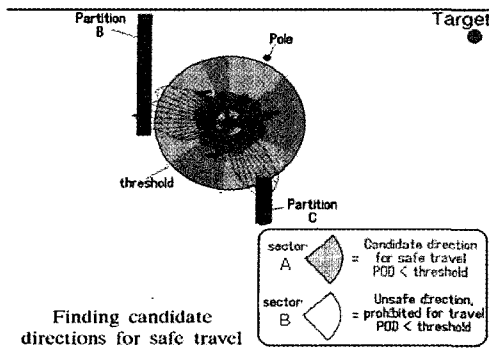


그림 14. 극 히스토그램의 기준 영역
Fig. 14. Threshold of Polar histogram

다음은 장애물의 위치와 장애물과의 거리에 따라 어떻게 로봇의 진행 방향이 결정 되는지 보여 주며, 그림 15로부터 17 까지 장애물의 위치에 따라 로봇의 진행방향이 어느 방향으로 결정되는지 보여주고 있다.

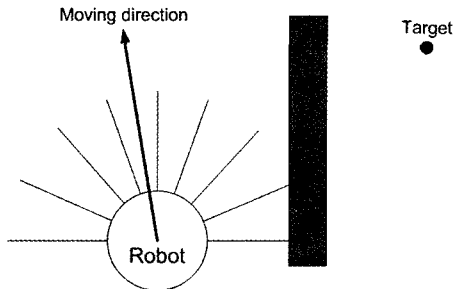


그림 15. 로봇이 장애물에 가까운 경우
Fig. 15. Robot is too close to the obstacle

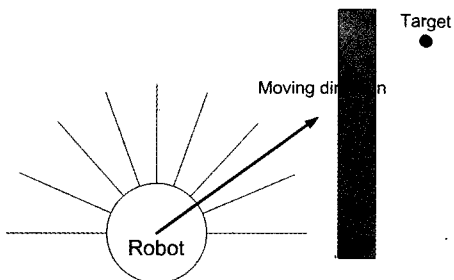


그림 16. 로봇이 장애물과 먼 경우
Fig. 16. Robot is far from the obstacle

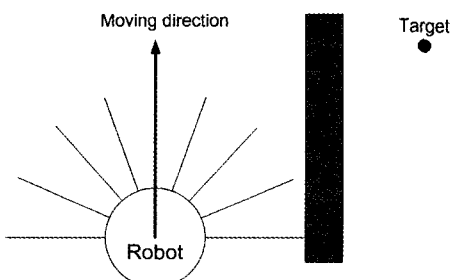


그림 17. 로봇이 장애물과 적절한 거리에 있는 경우
Fig. 17. Robot is at the proper distance from the obstacle

본 논문에서 구현한 로봇은 9개의 센서를 180°에 걸쳐서 가지고 있고, 로봇은 그림 15~17에서 보여 주는 것과 같이 장애물 회피 행위를 하게 된다. 센서의 범위를 기준 거리로 두고 시뮬레이션을 하였고, 각각의 경우에 로봇은 위와 같은 진행 방향을 가지게 된다.

5. Supervisor

본 논문에서 퍼지 알고리즘과 VFH 알고리즘 2가지를 사용하여 자율이동로봇을 수행시켰다. 각각의 알고리즘으로 주행 시뮬레이션을 하여 본 결과, 검출된 장애물의 형태에 따라서 알고리즘의 성능이 다름을 알 수 있었다. 즉, 벽면을 따라가는 경우 VFH 알고리즘이 더 뛰어난 성능을 보였고, 일반적인 장애물과 모서리와 같은 뾰족한 모양의 장애물 근처에서는 퍼지 알고리즘이 더 뛰어난 성능을 보였다. 그래서 본 논문에서는 초음파 센서를 통해 검출한 데이터를 바탕으로 장애물 환경을 파악하고, 해당 환경에서 더 우수한 성능을 갖는 알고리즘을 선택하는 상위레벨의 supervisor를 두어 전체적으로 로봇 성능이 향상 되도록 하였다.

그림 18은 supervisor에 사용된 가중치 함수이며, 장애물이 벽면일 경우 측면 센서를 통해 데이터가 들어올 확률이 크므로 측면 센서에 더 큰 가중치를 주었다. 식(17)에 의한 결과 값이 기준 값을 초과할 경우에 VFH 알고리즘을 선택하도록 하였으며, 기준 값 미만의 값이 나올 경우에 퍼지 알고리즘을 선택하도록 하였다.

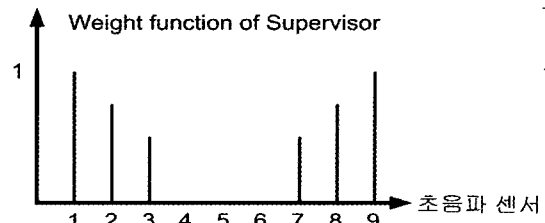


그림 18. Supervisor의 가중치 함수
Fig. 18. Weight function of supervisor

$$switching\ value = \frac{\sum_{i=1}^9 sensor\ value(x) * weight(x)}{number\ of\ detected\ sensors} \quad (17)$$

6. 모의실험 및 결과

모의실험 작업 공간의 크기는 400[cm] * 400[cm]이고 자율이동로봇에 대한 설정치는 표 5와 같으며 시뮬레이션 환경은 그림 19와 같다.

표 5. 모의실험에 사용된 로봇의 설정치
Table 5. Specification of robot in simulation environments

로봇 직경	40[cm]
샘플링 주기	0.01[sec]
센서 측정 거리	50[cm]
로봇의 최대 속도	40[cm/s]

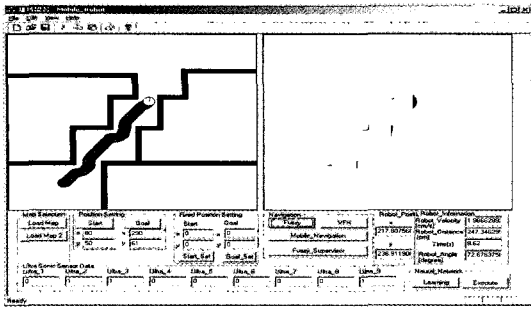


그림 19. 시뮬레이션 환경
Fig. 19. Simulation environment

본 연구에서는 자율이동로봇이 목표점까지 안전하게 주행할 수 있도록 퍼지 알고리즘과 VFH 알고리즘을 로봇에 적용시켜 각각 구현해 보았으며, 환경에 따라 더 뛰어난 알고리즘을 선택하여 자율이동로봇의 주행에 적용시켜 주는 supervisor를 만들고 위의 두 가지 알고리즘을 조합하는 방법도 구현하였다.

6.1 장애물이 없는 경우

그림 20은 장애물이 없는 경우에 퍼지 알고리즘을 사용하여 주행한 결과이다. 목표점까지의 소요 시간은 10.36초이다.

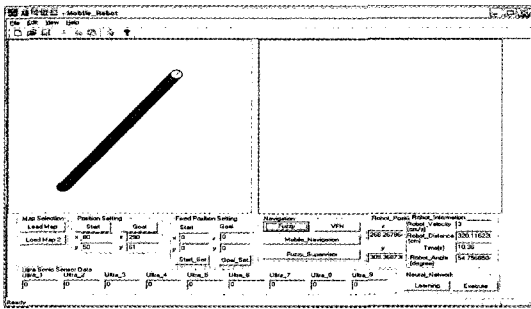


그림 20. 장애물이 없는 경우 퍼지 알고리즘 적용 결과
Fig. 20. Simulation results for the fuzzy algorithm without obstacle

그림 21은 장애물이 없는 맵에서 VFH 알고리즘을 사용하여 주행한 결과이다. 목표점까지의 소요 시간은 10.65초이다.

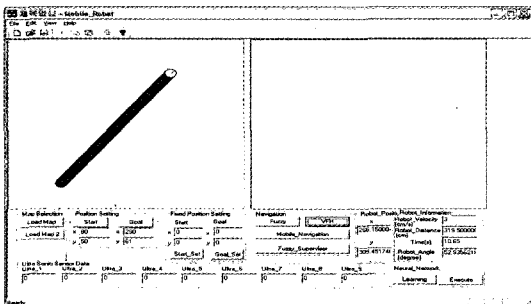


그림 21. 장애물이 없는 경우 VFH 알고리즘 적용 결과
Fig. 21. Simulation results for VFH algorithm without obstacle

즉, 장애물이 없는 경우 퍼지 알고리즘이 VFH 알고리즘보다 더 빠른 시간 안에 목표점에 도착하는 성능을 보인다.

6.2 모서리가 많은 장애물의 경우

그림 22는 모서리가 많은 장애물이 있는 경우, 퍼지 알고리즘을 사용하여 주행한 결과이며 목표점까지 소요시간은 12.4초이다.

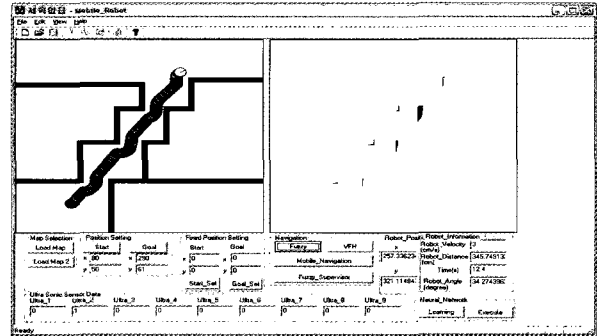


그림 22. 모서리 장애물에서 퍼지 알고리즘 적용 결과
Fig. 22. Simulation results for the fuzzy algorithm in sharp edge obstacles

그림 23은 모서리가 많은 장애물이 있는 경우, VFH 알고리즘을 사용하여 주행한 결과이며 목표점까지 소요시간은 12.83초이다.

모서리 장애물 환경에서도 퍼지 알고리즘이 VFH 알고리즘보다 더 우수한 성능을 보이는 것을 알 수 있다.

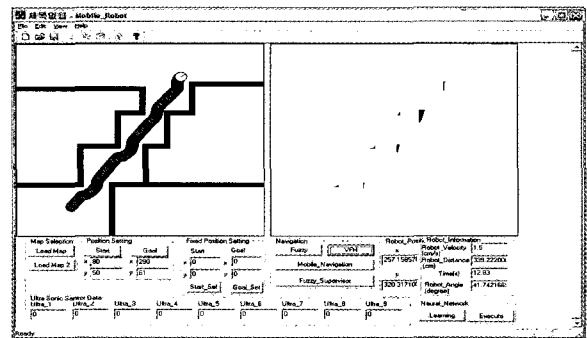


그림 23. 모서리 장애물에서 VFH 알고리즘 적용 결과
Fig. 23. Simulation results for VFH algorithm in sharp edge obstacles

6.3 벽면을 따라가는 경우

그림 24는 벽면을 따라갈 경우, 퍼지 알고리즘을 사용하여 주행한 결과이며 목표점까지 소요시간은 12.11초이다. 그림 25는 벽면을 따라갈 경우, VFH 알고리즘을 사용하여 주행한 결과이며 목표점까지 소요시간은 11.94초이다.

로봇이 벽면을 따라갈 때에 VFH 알고리즘이 퍼지 알고리즘보다 더 우수한 성능을 보이는 것을 알 수 있다.

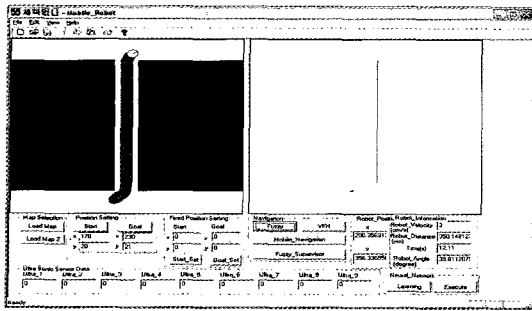


그림 24. 벽면에서 퍼지 알고리즘 적용 결과
Fig. 24. Simulation results for the fuzzy algorithm in hallway

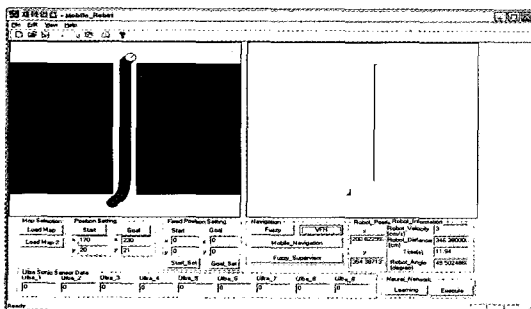


그림 25. 벽면에서 VFH 알고리즘 적용 결과
Fig. 25. Simulation results for VFH algorithm in hallway

6.4 일반적인 환경의 경우

위와 같이 장애물이 없거나 장애물의 모서리를 주행하는 경우에 퍼지 알고리즘이 더 뛰어난 성능을 보였으며, 벽면을 따라서 주행 하는 경우에 VFH 알고리즘이 더 뛰어난 성능을 보였다. 이 정보를 바탕으로 상위 레벨의 supervisor가 장애물의 형태에 따라 더 뛰어난 성능의 알고리즘을 선택하도록 조합 주행 알고리즘을 구성하고, 이 조합 주행 알고리즘으로 로봇을 구동하였다.

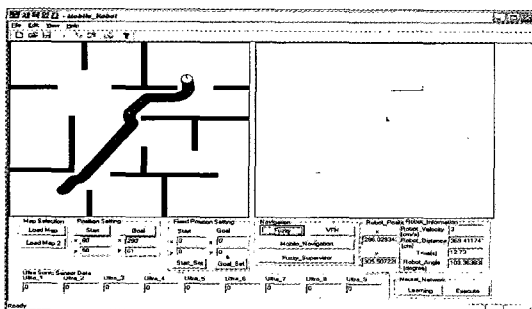


그림 26. 일반 환경 1에 퍼지 알고리즘 적용 결과
Fig. 26. Simulation results for the fuzzy algorithm in general environment 1

벽면과 모서리가 공존하는 일반적인 환경을 형성하고, 그 환경에서 자율이동로봇 주행을 시뮬레이션 하였으며, 퍼지 알고리즘과 VFH 알고리즘을 각각 사용한 경우와 supervisor에 의한 조합 주행 알고리즘을 사용한 경우의 성능들을 비교하였다. 그림 26과 27은 일반 환경 1과 2에서 퍼지 알고리즘을 사용하여 로봇을 주행시킨 결과이다. 목표점까지 주

행 소요시간은 각각 12.73초, 14.72초 이다.

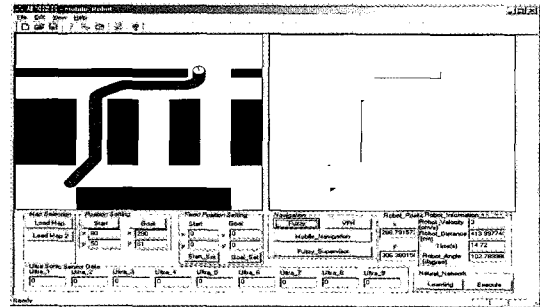


그림 27. 일반 환경 2에 퍼지 알고리즘 적용 결과
Fig. 27. Simulation results for the fuzzy algorithm in general environment 2

그림 28과 29는 일반 환경 1과 2에서 VFH 알고리즘을 사용하여 로봇을 주행시킨 결과이다. 목표점까지 소요시간은 각각 13.81초, 14.32초 이다.

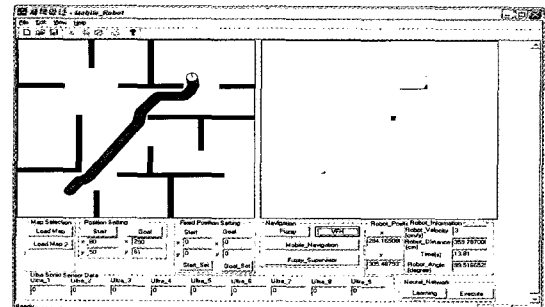


그림 28. 일반 환경 1에 VFH 알고리즘 적용 결과
Fig. 28. Simulation results for VFH algorithm in general environment 1

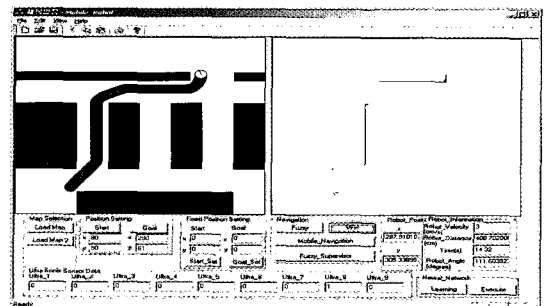


그림 29. 일반 환경 2에 VFH 알고리즘 적용 결과
Fig. 29. Simulation results for VFH algorithm in general environment 2

그림 30과 31은 일반 환경 1과 2에서 supervisor로 퍼지 알고리즘과 VFH 알고리즘 2개를 조합한 경우의 로봇주행 시뮬레이션 결과이다. 목표점까지 소요시간은 각각 11.99초, 13.51초 이다.

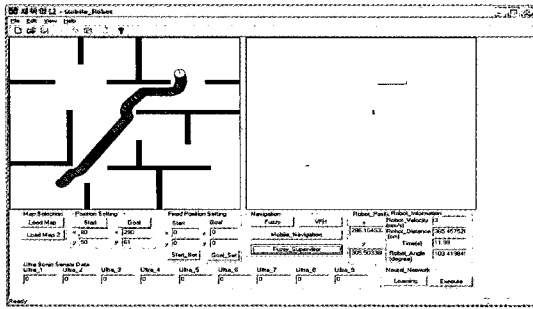


그림 30. 일반 환경 1에 조합 주행 알고리즘 적용 결과
Fig. 30. Simulation results for the combined navigation algorithm in general environment 1

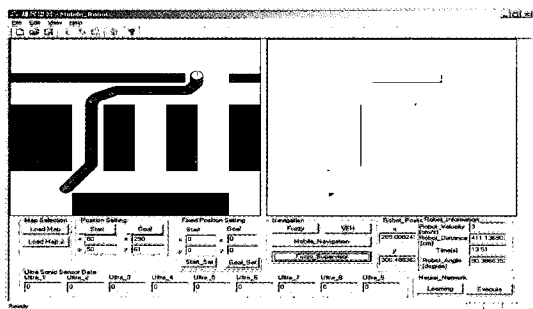


그림 31. 일반 환경 2에 조합 주행 알고리즘 적용 결과
Fig. 31. Simulation results for the combined navigation algorithm in general environment 2

Supervisor로 조합한 주행 알고리즘을 사용한 경우가 그렇지 않은 경우들보다 우수한 성능을 보임을 알 수 있다. 표 6은 이상의 모의실험 결과들을 정리한 것이다.

표 6. 모의실험 결과

Table 6. Results of simulation experiments

	퍼지 알고리즘	VFH 알고리즘	조합 주행 알고리즘
장애물이 없는 경우	10.36초	10.65초	
모서리가 많은 장애물의 경우	12.40초	12.83초	
벽면을 따라가는 경우	12.11초	11.94초	
일반 환경 1	12.73초	13.81초	11.99초
일반 환경 2	14.72초	14.32초	13.51초

7. 결 론

자율이동로봇이 주위 환경을 알지 못하는 상황에서 목표점까지 안전하게 주행하게 하는 주행 알고리즘에 대해 연구하였다. 이를 위해 장애물 회피 알고리즘인 VFH 알고리즘과 퍼지 알고리즘을 여러 가지 환경에서 비교 분석하였는데, 장애물이 없거나 장애물의 모서리를 주행하는 경우에 퍼지 알고리즘이 더 뛰어난 성능을 보였고 벽면을 따라서 주행하는 경우에 VFH 알고리즘이 더 뛰어난 성능을 보였다. 이 정보를 바탕으로 상위 레벨의 supervisor가 장애물의 형태에 따

라 더 뛰어난 성능의 알고리즘을 선택하도록 조합 주행 알고리즘을 구성하고, 이 조합 주행 알고리즘으로 로봇을 구동하여 여러 가지 형태의 환경에서 최적의 성능을 내도록 하였다. 다양한 환경에서 모의실험을 수행함으로써 제안된 로봇 주행 알고리즘의 우수성을 확인하였다.

참 고 문 헌

- [1] Y. Kanayama, "A Locomotion Method for Autonomous Vehicle," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1315-1317, 1998.
- [2] A. Elfes, "Sonar-based Real-world Mapping and Navigation," IEEE Journal of Robotics and Automation, Vol. 3, No. 4, pp. 249-265, June 1987.
- [3] A. Zelinsky, "A Mobile Robot Exploration Algorithm," IEEE Transactions on Robotics and Automation, Vol. 8, No. 6, pp. 707-717, Dec. 1992.
- [4] J. Borenstein and Y. Koren, "Obstacle Avoidance with Ultrasonic Sensors," IEEE Journal of Robotics and Automation, Vol. 4, No. 2, pp. 213-218, 1988.
- [5] W. Li, "Fuzzy Logic based Robot Navigation in Uncertain Environments by Multisensor Integration," Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 259-264, 1994.
- [6] K. Pal Prabir and Kar Asim, "Mobile Robot Navigation using A Neural Net," Proceedings of the IEEE International Conference on Robotics and Automation, Vol. 2, pp. 1503-1508, 1995.
- [7] J. Borenstein and Y. Koren, "The Vector Field Histogram- Fast Obstacle Avoidance for Mobile Robots," IEEE Transactions on Robotics and Automation, Vol. 7, No. 3, pp. 278-288, June 1991.
- [8] Shigeki Ishikawa, "A Method of Indoor Mobile Robot Navigation by Using Fuzzy Control," Proceedings of the IEEE International Conference on Intelligent Robots and Systems, vol. 2, pp. 1013-1018, 1991.
- [9] E. Tunstel, H. Danny, T. Lippincott and M. Jamshidi, "Autonomous Navigation Using An Adaptive Hierarchy of Multiple Fuzzy-behaviors," Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp. 276-281, 1997.
- [10] G. Dudek and M. Jenkin, Computational Principles of Mobile Robotics, Cambridge University Press, 2000.
- [11] E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with A Fuzzy Logic Controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.

저 자 소 개



이헌재 (Hyun-Jae Yi)

1979년 : 9월 7일생.

2003년 : 동아대 공대 기계공학과 졸업.

2006년 : 부산대 전기공학과 졸업(석사).

Phone : 019-533-8086
Fax : 051-513-0212
E-mail : waryong@pusan.ac.kr



최영규 (Young-Kiu Choi)

1957년 : 6월 18일생.

1980년 : 서울대 공대 전기공학과 졸업.

1982년 : 한국과학기술원 전기 및 전자공학과 졸업(석사).

1987년 : 동 대학원 전기 및 전자공학과 졸업(박사).

1990~1991년 : CALTECH Visiting Scholar.

1998~1999년 : Univ. of Southwestern Louisiana Visiting Professor.

현재 : 부산대 공대 전자전기통신공학부 교수

Phone : 051-510-2371
Fax : 051-513-0212
E-mail : ykichoi@pusan.ac.kr