

# 모바일 로봇의 주행 능력 향상을 위한 이중 룰 평가 구조의 퍼지 기반 자율 주행 알고리즘

박 기 원<sup>†</sup>

## Fuzzy Logic Based Auto Navigation System Using Dual Rule Evaluation Structure for Improving Driving Ability of a Mobile Robot

Kiwon Park<sup>†</sup>

### ABSTRACT

A fuzzy logic based mobile robot navigation system was developed to improve the driving ability without trapping inside obstacles in complex terrains, which is one of the most concerns in robot navigation in unknown terrains. The navigation system utilizes the data from ultrasonic sensors to recognize the distances from obstacles and the position information from a GPS sensor. The fuzzy navigation system has two groups of behavior rules, and the robot chooses one of them based on the information from sensors while navigating for the targets. In plain terrains the robot with the proposed algorithm uses one rule group consisting of behavior rules for avoiding obstacle, target steering, and following edge of obstacle. Once trap is detected the robot uses the other rule group consisting of behavior rules strengthened for following edge of obstacle. The output signals from navigation system control the speed of two wheels of the robot through the fuzzy logic data process. The test was conducted in the Matlab based mobile robot simulator developed in this study, and the results show that escaping ability from obstacle is improved.

**Key words:** Robot Navigation, Fuzzy Systems, Fuzzy Logic Controller

### 1. 서 론

무인자율주행 로봇이 미래 주요 운송수단으로 부상하고 있는 가운데 효율적인 주행 알고리즘을 개발하기 위한 연구가 활발하게 진행되고 있다. 그 중에서 로봇의 자율주행에 보편적으로 사용되고 있는 알고리즘으로서 행위기반 주행 알고리즘(Behavior based navigation algorithm)이 있다 [1]. 행위기반 주행 알고리즘은 로봇에 부착된 센서들로부터 실시간으로 전해지는 다양한 외부 환경정보들에 대응하여 사전에 프로그램 되어있는 행위 규칙들(Behavior

rules)이 로봇의 움직임을 결정하게 하는 주행 알고리즘이다. 이러한 행위기반 주행 알고리즘은 로봇의 움직임이 정해진 주행 경로를 따라가도록 사전에 결정되어 있지 않고 실시간으로 변하는 외부 환경에 대응하여 로봇 스스로가 움직임을 결정하도록 만들기 때문에 로봇의 장애물 회피 능력, 돌발 상황 대처 능력 및 목표 도달 능력을 향상시킬 수 있다. 그러나 행위기반 주행 알고리즘의 효율성을 더욱 향상시키기 위해서는 탑재된 센서들로부터 전달되는 외부 환경 정보들의 수많은 경우의 수에 대응하여 로봇의 움직임을 결정하는 행동 명령어들로 구성된 다수의

\* Corresponding Author : Kiwon Park, Address: Yangsan Campus 626-790, 288 Junam-ro, Yangsan, Gyeongnam, Korea, TEL : +82-55-380-9491, FAX : +82-55-380-9249, E-mail : kwp@ysu.ac.kr

Receipt date : Dec. 9, 2014, Revision date : Jan. 17, 2015

Approval date : Feb. 10, 2015

<sup>†</sup> Dept. of Green Automobile Engineering, School of Engineering Youngsan University

\* This work was supported by Youngsan University Research Fund of 2014.

행위 규칙들(Behavior rules)을 고안해야 하며 보다 자연스러운 주행을 위해서 규칙들 간에 결합 문제 등이 고려되어야 한다 [2].

퍼지 로직은 인간의 사물 인지 능력을 모방한 이론이다. 운전자가 운전 중에 주위 환경으로부터 전달되는 많은 정보들을 조합하여 탁월한 주행능력을 발휘하듯이 인간의 인지능력을 로봇의 자율 주행에 접목시킬 경우 주변 변수들의 처리 능력 및 이에 대한 대처 능력을 향상시킬 수 있어서 자율 주행 알고리즘 개발에 퍼지 로직은 널리 응용되고 있다[2-6]. 퍼지 로직은 퍼지화(Fuzzification), 룰평가(Rule evaluation), 비퍼지화(Defuzzification)의 3단계의 절차를 통하여 출력 신호를 생성한다. 퍼지화 단계에서는 퍼지 멤버 함수들을 사용하여 센서 데이터의 가중치가 결정된다. 룰 평가단계에서는 퍼지화된 데이터의 가중치에 따라 해당되는 행위 규칙들이 결정이 되며, 비퍼지화 단계에서는 행위 규칙들의 출력을 조합하고 비퍼지화 함수를 거치면서 로봇의 구동 신호가 생산되게 된다. 퍼지로직 알고리즘의 단계별 구성이 가지는 특성으로 인하여 입력 신호, 행위 규칙, 출력 신호간의 연결이 자연스럽게 이루어지기 때문에 행위 규칙의 설계 및 규칙간의 결합 문제를 보다 효율적으로 해결 할 수 있을 뿐만 아니라 모든 입력 신호들이 가중치가 고려되어 출력에 영향을 미치기 때문에 로봇 주행 알고리즘과 같이 불특정하고 다수의 환경적 변수들을 다루는 시스템에서 탁월한 성능을 발휘하게 된다[2-6].

퍼지 로직을 사용하는 행위기반 주행 알고리즘이 로봇이 자율 주행에 있어서 가지는 한가지 문제점은 로봇이 장애물 속에 갇혀 벗어나지 못하는 경우가 발생하는 점이다[1,6]. 이 문제는 행위 규칙들 중에서 로봇의 목표지점 추종을 위한 행위 규칙들과 장애물 회피를 위한 행위 규칙들이 함께 작용하면서 같은 지점을 계속 맴돌게 만드는 특정 형상의 장애물들 속에서 발생하게 된다. 로봇의 갇힘 현상을 야기하는 대표적인 장애물로서 U자 형상의 장애물이 있다. Fig. 1은 자동 주행 로봇이 목표지점에 도달하지 못하고 U자 형상의 장애물 속에 갇히게 되는 대표적인 경우를 묘사하고 있다. 로봇이 목표지점을 향해 움직이면서 장애물을 만나게 되면 장애물 회피를 위한 행위 규칙들이 작동하게 되고 로봇은 B 지점까지 움직이게 된다. B 지점에서는 다시 목표지점 추종을

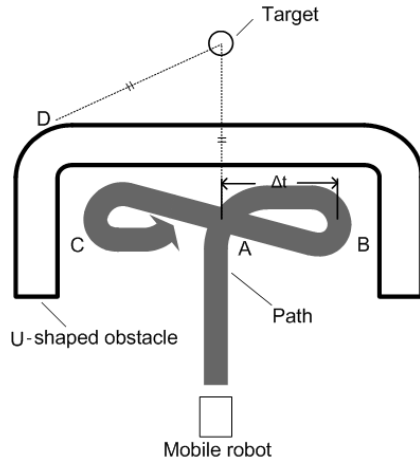


Fig. 1. Description of navigation failure in an obstacle.

위한 규칙들과 장애물 회피를 위한 규칙들에 의해서 A 지점을 지나 C 지점으로 이동하게 된다. C 지점에서는 다시 목표지점 추종을 위한 규칙들과 장애물 회피를 위한 규칙들에 의해서 A 지점으로 로봇이 이동하게 되며, 이와 같은 현상이 계속 반복이 되기 때문에 장애물의 내부를 벗어나지 못하고 그 곳에 갇히게 된다. 따라서 로봇이 복잡한 환경 속에서 주행 중 장애물 사이에 갇히지 않고 차례로 지정된 목표 지점을 거쳐 최종 목표지점에 도달 할 수 있기 위해서는 장애물 속에서 탈출구를 찾아 빠져 나올 수 있는 보다 향상된 주행 기술의 고안이 요구된다.

U자 형상의 장애물 속에 갇히게 되는 현상은 다른 주행 알고리즘들을 사용한 로봇 주행에서도 나타나는 현상이다[7-10]. 대표적인 로봇 주행 알고리즘에는 Bug 알고리즘, Potential field 알고리즘, Vector field histogram 알고리즘 등이 있다. Bug 알고리즘은 이들 중에서 가장 먼저 개발된 알고리즘으로써 Bug 알고리즘을 탑재한 로봇은 주행 중에 장애물을 만날 경우 장애물의 가장자리를 따라 일주하게 된다 [7,8]. 이 과정에서 찾아낸 목표지점과의 최단 직선거리 지점을 탈출 지점으로 지정하고 다시 그 지점으로 이동 후에 목표지점을 향해 움직이게 된다. Bug 알고리즘은 모든 장애물을 일주하면서 탈출 지점을 찾기 때문에 주행 도중 U자형 장애물을 만날 경우 탈출 지점을 찾기까지 많은 시간이 소요되는 단점을 가진다. Potential field 알고리즘에서는 로봇과 장애물들을 양의 전위를 가지는 대상으로 간주하고 목표지점을 음의 전위를 가지는 대상으로 정의한다. 따라서

로봇과 장애물들 사이에는 척력이 그리고 로봇과 목표지점 사이에는 인력이 작용하게 되며, 인력은 로봇과 목표지점사이의 거리에 비례한다고 정의한다[7, 9,10]. 이로부터 주행 환경 상에서 로봇에게 작용하는 힘의 벡터로 구성된 potential field가 형성되며, 로봇은 작용하는 힘이 이끌리어 목표지점을 향해서 주행하게 된다. 그러나 Potential field를 계산하기 위해서는 주행 지형에 대한 전반적인 사전 정보가 필요하며 로봇이 지형에 대한 사전 정보 없이 주행 할 경우에는 적용하기 힘든 알고리즘이다. 또한 U자형 장애물의 내부는 인력과 척력이 평형상태가 되는 힘이 0이 되는 국소 최소점이 존재하기 때문에 로봇이 장애물 내부에서 정지하는 현상이 발생하게 된다. Vector field histogram 알고리즘의 경우 로봇의 회전 각도에 대응하여 장애물로부터의 거리를 나타내는 histogram의 밀집도가 낮은 지점으로 로봇이 이동하게 된다[9]. 그러나 U자형 장애물의 내부에서는 목표지점 추종을 위한 주행 알고리즘의 영향으로 histogram에서 나타내는 탈출 지점을 벗어나지 못하고 장애물의 내부를 맴돌게 된다. 따라서 U자 형상의 장애물 내부에 갇히는 문제는 로봇이 사전에 지리적 정보에 기반을 둔 주행 시나리오를 가지고 주행을 할 경우 혹은 탑재된 비전 센서로 부터의 시각적 정보를 활용하는 등의 경우 [1,3,6,11-13]들을 제외하고는 여전히 풀어야 할 과제로 남아 있는 실정이다.

본 연구에서는 로봇이 주행 중에 U자형 장애물의 내부에 갇히는 문제를 해결하기 위해서 이중 구조의 규칙평가 단계를 가지는 퍼지로지 기반의 주행 알고리즘을 설계하고 그 성능을 단일 규칙평가 단계를 가지는 일반적인 퍼지 주행알고리즘의 성능과 비교하였다. 시뮬레이션 결과 제안된 구조의 퍼지 알고리즘을 사용하여 주행 중에 장애물에 갇힐 경우 탈출 능력이 향상되었음이 검증되었다.

본 논문의 구성은 다음과 같다. 2장에서 Matlab 기반의 시뮬레이션 환경에서 구현된 모바일 로봇 상에 탑재된 센서들의 배치 및 각각의 역할에 관해서 설명한 다음 본 논문에서 사용된 퍼지 주행 알고리즘에 관한 설명과 센서들로부터 얻어진 데이터들이 퍼지 알고리즘을 거쳐 로봇의 이동 방향 및 속도를 결정하는 과정을 설명한다. 여기에서는 제안한 퍼지 알고리즘에서 사용된 퍼지 멤버 함수들을 사용하는 퍼지화 단계 그리고 로봇의 움직임을 결정하는 행위 유닛들

을 사용하는 퍼지 규칙평가 단계 및 실제 모터의 회전 속도를 결정하는 비퍼지화 단계에 관해서 서술한다. 또한 본 연구에서 개발된 로봇 시뮬레이터의 설계 과정을 소개한다. 3장에서는 다양한 시뮬레이션 환경을 구성하고 그 속에서 제안한 퍼지 주행 알고리즘을 사용하는 모바일 로봇의 이동 경로를 시각적으로 평가하여 퍼지 주행알고리즘의 성능을 검증하였다. 마지막으로 본 연구의 결론을 제시한다.

## 2. 퍼지 주행 알고리즘

### 2.1 퍼지 주행 알고리즘의 입력 신호

본 논문에서는 외부환경 속에서 로봇의 주행성능이 평가되었으며, 8개의 거리 센서와 1개의 GPS 센서가 로봇에 탑재되어 사용되었다. 이 경우 로봇은 사전 지리적 정보 없이 목표 지점을 찾아 운행하는 동안에 탑재된 거리 센서들에 의해서 주변 장애물들을 인지하게 된다. 또한 로봇은 자신의 현재 위치를 탑재된 GPS 센서에 의해서 실시간으로 인지하게 된다. Fig. 2(a)는 시뮬레이션 환경에서 구현된 거리센서의 기능을 도식적으로 보여준다. 거리센서는 로봇의 전면에 2개 왼쪽과 오른쪽에 각각 3개씩 탑재되어 있다. 총 8개의 거리센서들로부터 얻어지는 데이터 중 퍼지 주행 알고리즘으로 전달되는 입력 신호는 각면의 센서들의 값 중에서 최소값, 즉  $f_d = \min\{d_1, d_2\}$ ,  $r_d = \min\{d_3, d_4, d_5\}$ ,  $l_d = \min\{d_6, d_7, d_8\}$ 로 구성된 3가지 값들이 퍼지 주행 알고리즘의 입력들로 사용된다.

Fig. 2(b)는 로봇에 탑재된 GPS 센서의 기능을 도식적으로 보여준다. GPS 센서는 로봇의 상부 중앙에 위치하며 센서로부터 얻어진 현재 로봇의 위치 좌표 값을 사용하여 로봇의 현재 진행 방향과 현재 위치에서 목표지점까지의 직선거리가 만들어 내는 각도의 차이 ( $\theta_d$ )를 계산하여 퍼지 주행 알고리즘의 입력으로 사용한다. 따라서 주행 알고리즘의 입력의 개수는 총 4가지이며 각각의 변수 명은  $f_d, r_d, l_d, \theta_d$ 이다.

### 2.2 이중 규칙평가 구조의 퍼지 주행 알고리즘

모바일 로봇이 목표 지점에 도달하기 위해서는 목표지점 추종, 장애물 회피, 장애물의 가장자리 따라가기와 같은 주행 기술들이 필요하며, 이러한 주행기

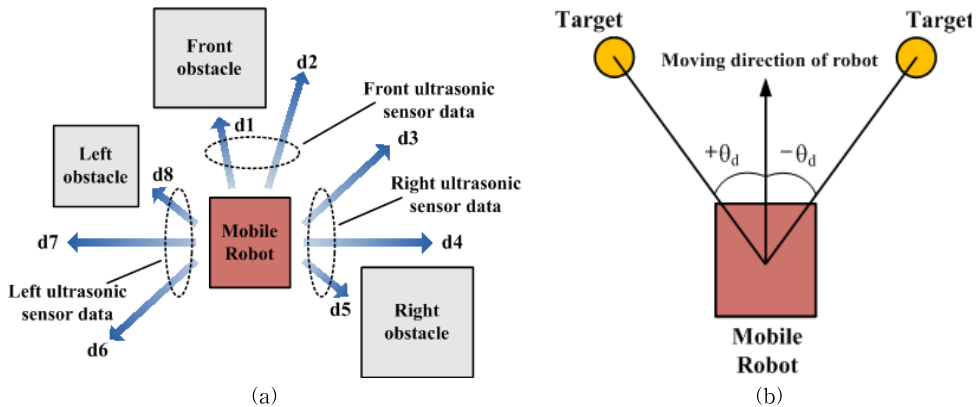


Fig. 2. Description of sensors mounted on the mobile robot: (a) Performance description of distance sensors, (b) Performance description of GPS sensor.

술들의 효율성은 여러 연구들에 의해서 입증되고 있다[14-19]. 그러나 일반적인 주행 기술만으로는 로봇이 장애물에 간헐하게 되는 문제를 해결하기 어렵고 결국에는 주행 실패의 결과를 야기한다.

본 논문에서는 모바일 로봇의 주행 상에 발생하는 장애물 속 갇힘 문제를 해결하기 위하여 이중 규칙평가 구조를 가지는 퍼지 주행 알고리즘을 설계하였다. Fig. 3은 본 논문에서 제안한 퍼지 주행 시스템의 구조를 보여준다.

Fig. 3에서 제시된 주행 알고리즘을 탑재한 로봇은 일반적인 지형을 운행 중에는 Rule group A를 사용하여 목적지에 도착하게 된다. 그러나 장애물 속에 갇혔음이 감지될 경우 장애물로 부터의 탈출구를 찾을 때 까지 Rule group B를 사용하여 장애물의 가장자리를 따라가게 된다. 따라서 로봇이 Rule group B를 사용하는 동안에는 GPS 센서로부터 영향을 받지 않고 장애물과 충돌하지 않는 범위에서 장애물의 가장자리를 따라 이동하게 된다.

$$x_{RMS} = \sqrt{\frac{\text{mean}\{x[1], x[2], \dots, x[n]\} - \text{mean}\{x[1+N-n], x[2+N-n], \dots, x[N]\}}{n}}^2 \quad (1)$$

$$y_{RMS} = \sqrt{\frac{\text{mean}\{y[1], y[2], \dots, y[n]\} - \text{mean}\{y[1+N-n], y[2+N-n], \dots, y[N]\}}{n}}^2 \quad (2)$$

Fig. 3의 Rule Decision 단계에서 주행 알고리즘은 센서들로부터 얻어진 데이터를 분석하여 장애물 속에 갇혔음을 판단하게 된다. Rule group A를 사용하여 로봇이 주행하는 중에 장애물 속에 갇히게 되면 Fig. 1에 묘사된 것과 같이 동일 지점을 맴돌게 되는데, 이러한 주행 패턴 상에서 발생하는 현상을 감지

하기 위해 Rule Decision 단계에서는 현재 지점 부근에서 얻어진 x, y 위치 좌표 정보들의 개별 평균값과 Δt 이전 지점 부근에서 얻어진 x, y 위치 좌표들의 평균값들 사이의 RMS(Root Mean Square) 에러 값들( $x_{RMS}, y_{RMS}$ )이 모두 사전 정의된  $d_1$ 과  $d_2$ 의 크기보다 작을 경우 로봇이 유사 지점을 지나가는 것이므로 특정 지역을 벗어나지 못하고 갇힌 것으로 간주하여 Rule group B를 작동시키게 된다. 갇힘 감지 알고리즘의 구현을 위해 주행 중에 실시간으로 전달되는

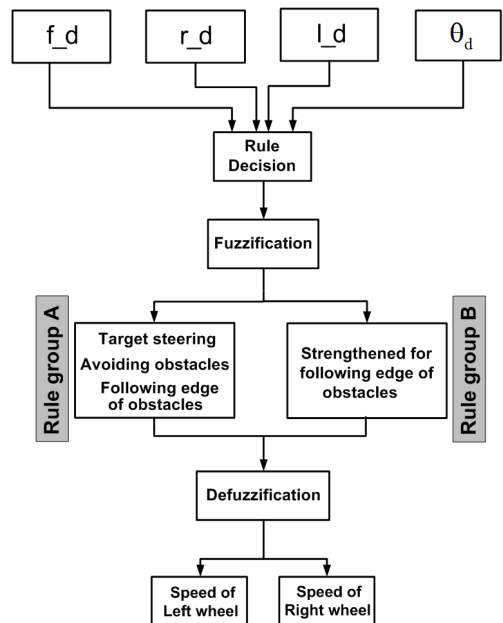


Fig. 3. The scheme of navigation system with dual rule evaluation structure.

좌표 데이터를 저장하여 현재 시점부터  $N$ 개만큼의 이전 좌표 데이터를 저장하도록 설계되었으며, 시간 순으로 저장된 데이터 값들 가운데 맨 앞과 뒤에서 각각  $n$ 개 데이터의 평균값을 RMS 에러 계산을 위해 사용하였다. 여기서  $N$ 은 GPS 센서 데이터의 샘플링 주파수와  $\Delta t$ 의 곱으로 정의되며, RMS 에러는 식 (1) 과 (2)로부터 계산되어 진다. Fig. 4는 Rule Decision 단계의 절차를 도식적으로 보여준다.

Rule group B가 선택되게 되면 로봇은 탈출지점을 발견할 때까지 장애물의 가장자리를 계속해서 따라가게 된다. 이 과정에서 로봇과 목표지점 사이의 직선거리가 장애물 속에 갇힘이 감지된 지점 (Fig. 1에서 A지점)에서의 로봇과 목표지점 사이의 거리 보다 작은 지점에 로봇이 도달했을 경우 (Fig. 1에서 D 지점을 넘어설 경우)를 탈출지점으로 간주하고 다시 Rule group A를 사용하게 되며 다른 갇힘이 감지되지 않는 한 장애물들을 회피하며 목표지점으로 이동하게 된다.

2.3 퍼지화(Fuzzification)

퍼지 주행 알고리즘에서 센서로부터 들어온 4가지 입력 신호들은 퍼지화 단계를 거치게 된다. 퍼지화 단계에서는 샘플링 과정을 거친 입력 신호 값들마다 고유의 멤버 함수들을 사용하여 가중치를 가지는 값들로 재분류된다.

Fig. 5(a)는 거리 센서들로부터 들어오는 입력신호들  $f_d, r_d, L_d$ 의 퍼지화에 사용된 멤버 함수이며, Fig. 5(b)는 GPS 센서로부터 들어오는 입력신호  $\theta_a$ 의 퍼지화에 사용된 멤버 함수이다. 입력  $f_d, r_d, L_d$ 는 멤버 함수를 사용한 퍼지화 과정을 거치면서 각각 Close, Med, Far의 이름들을 가지는 신호들로 재분류되며 각각 0에서 1사이의 가중치를 가지게 된다. 또한 입력  $\theta_a$ 는 Fig. 5(b)의 멤버 함수를 사용한다.

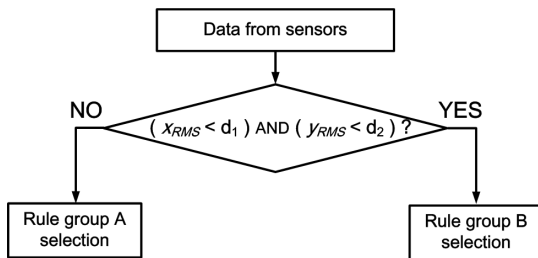


Fig. 4. Description of rule decision process.

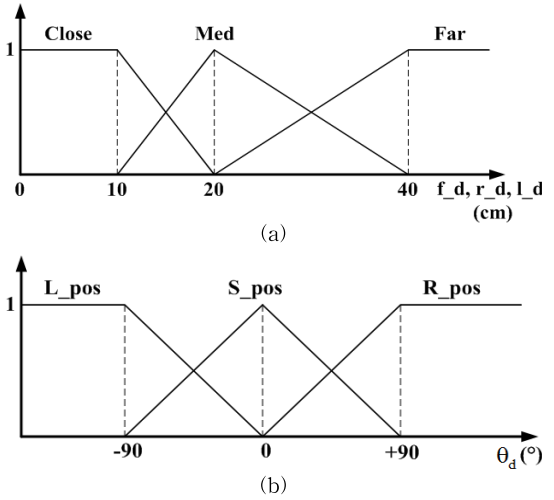


Fig. 5. Input membership functions: (a) Membership function for the distance data, (b) Membership function for the GPS data.

퍼지화 과정을 거치면서  $L\_pos, S\_pos, R\_pos$ 의 이름들을 가지는 신호들로 분류되며 각각 0에서 1사이의 가중치를 가지게 된다. 예를 들어  $f_d=10, r_d=20, L_d=30, \theta_a=45^\circ$ 가 퍼지 알고리즘의 입력들일 경우 퍼지화 과정을 거치면서  $f_d=\{Close(1), Med(0), Far(0)\}, r_d=\{Close(0), Med(1), Far(0)\}, L_d=\{Close(0), Med(0.5), Far(0.5)\}$  그리고  $\theta_a=\{L\_pos(0), S\_pos(0.5), R\_pos(0.5)\}$ 와 같이 이름과 가중치를 가지는 신호들로 변환 된다.

입력 멤버 함수를 구성하는 각각의 이름들이 부여된 내부함수들의 범위는 주행 알고리즘이 센서 신호 데이터로부터 로봇과 장애물과의 거리 및 진행 방향을 판단하는 척도가 되기 때문에 멤버 함수의 범위 변화는 로봇의 움직임 변화에 영향을 준다. 예를 들어 거리 센서의 데이터를 퍼지화 하는 입력 멤버 함수에서 Close의 범위가 줄어들 경우 주행 알고리즘이 Close로 인식하는 범위가 줄어들기 때문에 주행 중에 로봇이 장애물에 더욱 근접하여 움직이게 된다. 본 연구에서 사용된 입력 멤버 함수의 범위들은 시뮬레이션 환경에서 다양한 결과들을 바탕으로 로봇이 최적의 주행 경로를 보일 수 있도록 정해졌다.

2.4 규칙평가(Rule Evaluation)

퍼지 주행 알고리즘의 입력 신호들은 퍼지화 단계를 거쳐 이름과 가중치를 가지는 신호들로 바뀌게

된다. 규칙평가 단계에서는 이 퍼지 신호들이 해당되는 행위유닛(Behavior unit) 규칙들에 적용이 되어 퍼지화된 출력 신호들을 생성하게 된다. 로봇의 움직임을 결정하는 규칙들은 퍼지화된 입력 신호들의 값에 따라 로봇의 두 바퀴의 속도를 제어할 수 있도록 if (f\_d and r\_d and l\_d and  $\Theta_d$ ) then(L\_vel and R\_vel)의 형식을 가지는 조건문들로 구성되어 있다. 여기서 L\_vel과 R\_vel은 로봇의 두 바퀴 속도를 결정하는 퍼지화된 출력 신호들을 나타내며 바퀴의 속도를 나타내는 Slow, Med, Fast의 3 단계로 구분되어 진다. 본 연구에서는 로봇의 자연스러운 주행을 위해 Rule group A 와 B에 각각 81개의 행위 규칙들이 사용되었으며 로봇의 모션을 제어하게 된다.

Rule group A 속한 행위 규칙들은 로봇에게 목표지점 추종, 장애물 회피, 장애물 가장자리를 따라가기와 같은 주행 기법을 제공한다. 예를 들어 로봇의 목표지점 추종을 위한 규칙들은 주변에 장애물들이 없는 상황에서 로봇이 목표지점을 향하여 움직일 수 있도록 Table 1과 같이 구성되어 있다.

또한 로봇이 주행 중에 장애물을 만났을 때 가장자리를 따라가면서 장애물 너머에 있는 목표지점을 찾아 진행할 수 있도록 구성된 규칙들은 Table 2와 같은 형식을 가진다.

로봇이 주행 중에 장애물을 만났을 때 장애물을 회피 할 수 있도록 구성된 장애물 회피 규칙들은

Table 3과 같은 형식을 가진다.

Rule group B 속한 행위 규칙들은 로봇이 탈출지점을 발견할 때까지 장애물의 가장자리만을 따라가도록 가장자리 따라가기 기능이 강화된 규칙들로 구성되어 있다. 주행중에 Rule group B가 선택되면, 로봇은 일정 간격을 유지하면서 장애물에 접근하게 되며 GPS 센서 데이터의 영향을 받지 않고 계속해서 가장자리를 따라가게 된다. Rule group B가 작동하는 동안에 로봇은 대부분의 구간에서 저속으로 움직이지만, 장애물의 코너를 따라 회전할 경우 양 측면에 부착된 센서가 장애물을 감지하지 못하여 로봇이 장애물을 놓치는 경우가 발생하기 때문에 상대적으로 속도를 높여 회전하도록 구성되어 있다. 로봇에게 이와 같은 주행 기능을 제공하는 Rule group B는 Table 4와 같은 형식을 가진다.

앞서 설명한 행위 규칙들을 사용하여 규칙평가 단계에서는 최종적으로 출력 변수 L\_vel과 R\_vel의 가중치가 결정되게 된다. 출력 변수들의 가중치는 If문 내부의 변수들(Close, Med, Far, S\_pos, L\_pos, R\_pos)의 가중치들 중에서 최소값으로 선정되기 때문에 If문 내부의 변수들 중에서 가중치가 단 한 개라도 0인 경우 해당되는 규칙은 출력 변수의 가중치 선정에서 제외된다. 이전 장의 퍼지화 설명에서 사용된 센서의 값들을 사용하여 Rule group A가 작동할 경우를 예로 들면 가중치가 0인 부분은 제외되기 때문

Table 1. Rules for target steering in robot navigation

Rule	<i>if</i>	L_d	<i>and</i>	f_d	<i>and</i>	r_d	<i>and</i>	$\Theta_d$	<i>then</i>	L_vel	<i>and</i>	R_vel
1		Far		Far		Far		S_pos		Fast		Fast
2	<i>if</i>	Far	<i>and</i>	Far	<i>and</i>	Far	<i>and</i>	L_pos	<i>then</i>	Fast	<i>and</i>	Slow
3		Far		Far		Far		R_pos		Slow		Fast

Table 2. Rules for following edge of obstacles in robot navigation

Rule	<i>if</i>	L_d	<i>and</i>	f_d	<i>and</i>	r_d	<i>and</i>	$\Theta_d$	<i>then</i>	L_vel	<i>and</i>	R_vel
4		Far		Far		Close		L_pos		Med		Med
5	<i>if</i>	Close	<i>and</i>	Far	<i>and</i>	Far	<i>and</i>	R_pos	<i>then</i>	Med	<i>and</i>	Med

Table 3. Rules for avoiding obstacles in robot navigation

Rule	<i>if</i>	L_d	<i>and</i>	f_d	<i>and</i>	r_d	<i>and</i>	$\Theta_d$	<i>then</i>	L_vel	<i>and</i>	R_vel
6		Med		Close		Close		any		Slow		Fast
7	<i>if</i>	Close	<i>and</i>	Close	<i>and</i>	Med	<i>and</i>	any	<i>then</i>	Fast	<i>and</i>	Slow
8		Close		Med		Close		any		Med		Med

Table 4. Rules for strengthened following edge of obstacles in robot navigation

Rule	<i>if</i>	L_d	<i>and</i>	f_d	<i>and</i>	r_d	<i>and</i>	$\Theta_d$	<i>then</i>	L_vel	<i>and</i>	R_vel
9	<i>if</i>	Far	<i>and</i>	Far	<i>and</i>	Med	<i>and</i>	any	<i>then</i>	Fast	<i>and</i>	Slow
10		Far		Far		Close		any		Med		Med
11		Far		Med		Med		any		Fast		Med
12		Far		Med		Close		any		Med		Med
13		Med		Med		Close		any		Med		Med
14	Med	Far	Close	any	Med	Med						

Table 5. Example of activated rules

Rule	<i>if</i>	L_d	<i>and</i>	f_d	<i>and</i>	r_d	<i>and</i>	$\Theta_d$	<i>then</i>	L_vel	<i>and</i>	R_vel
15	<i>if</i>	Med	<i>and</i>	Close	<i>and</i>	Med	<i>and</i>	S_pos	<i>then</i>	Fast	<i>and</i>	Slow
16		Med		Close		Med		R_pos		Slow		Fast
17		Far		Close		Med		S_pos		Slow		Fast
18		Far		Close		Med		R_pos		Slow		Fast

에  $f_d=\{Close(1)\}$ ,  $r_d=\{Med(1)\}$ ,  $L_d=\{Med(0.5), Far(0.5)\}$  그리고  $\Theta_d=\{S\_pos(0.5), R\_pos(0.5)\}$ 만이 해당되는 규칙들에게 적용이 된다. 따라서 이 경우에는 Rule group A의 81개 행위 규칙들 가운데서 Table 5에 있는 총 4개의 규칙들이 활성화된다.

위의 예로부터 최종적으로 도출되는 출력 변수들과 가중치들은 각각  $L\_vel=\{Fast(0.5), Slow(0.5), Slow(0.5), Slow(0.5)\}$ ,  $R\_vel=\{Slow(0.5), Fast(0.5), Fast(0.5), Fast(0.5)\}$ 이다.

2.5 비퍼지화(Defuzzification)

규칙평가 단계에서 도출된 퍼지화된 출력 변수들은 비퍼지화 단계에서 로봇의 속도를 결정하는 최종 출력 신호로 바뀌게 된다. 이 변환 과정은 비퍼지화 기법에 의해서 이루어지는데, 본 연구에서는 식 (3)로 정의되는 가중평균(Weighted Average) 기법이 사용되었다.

$$x = \frac{\sum_{i=1}^n \text{활성률의 개수} (m^i \times w^i)}{\sum_{i=1}^n \text{활성률의 개수} m^i} \tag{3}$$

여기서  $x$ 는 비퍼지화된 최종 출력 신호이며,  $m$ 은 규칙평가 단계에서 도출된 출력변수의 가중치들을 나타내며,  $w$ 는 비퍼지화 단계에서 출력신호를 구분 짓는 Slow, Med, Fast의 가중치들로서 출력 멤버함수에서 해당되는 영역의 중심 값을 나타낸다. Fig.

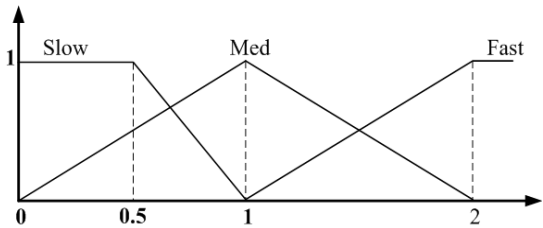


Fig. 6. Output membership function.

6은 본 연구에서 사용된 출력 멤버함수를 보여준다. 사용된 멤버함수의 경우 Slow, Med, Fast의 해당역역의 중심 값은 각각 0.5, 1, 2이다. 출력 멤버함수의 범위는 로봇의 이동 속도에 영향을 미치기 때문에 출력 멤버함수의 중심 값의 증감으로 로봇의 이동 속도를 증감시킬 수 있다.

규칙평가 단계에서 사용된 예를 적용할 경우에 계산된 퍼지 알고리즘의 최종 출력은 다음과 같다.

$$L\_vel = \frac{0.5 \times 0.5 + 0.5 \times 0.5 + 0.5 \times 0.5 + 0.5 \times 2}{0.5 + 0.5 + 0.5 + 0.5} = 0.88$$

$$R\_vel = \frac{0.5 \times 0.5 + 0.5 \times 2 + 0.5 \times 2 + 0.5 \times 2}{0.5 + 0.5 + 0.5 + 0.5} = 1.63$$

예시의 경우에는 로봇의 전면과 우측 가까이에 장애물이 감지되는 경우이기 때문에 계산 결과로부터 로봇이 왼쪽으로 방향을 틀수 있도록 오른쪽 바퀴의 회전 속력이 왼쪽 바퀴보다 빠르며, 따라서 퍼지 주행 알고리즘이 장애물을 회피하기 위해 적절하게 작동하고 있음을 알 수 있다.

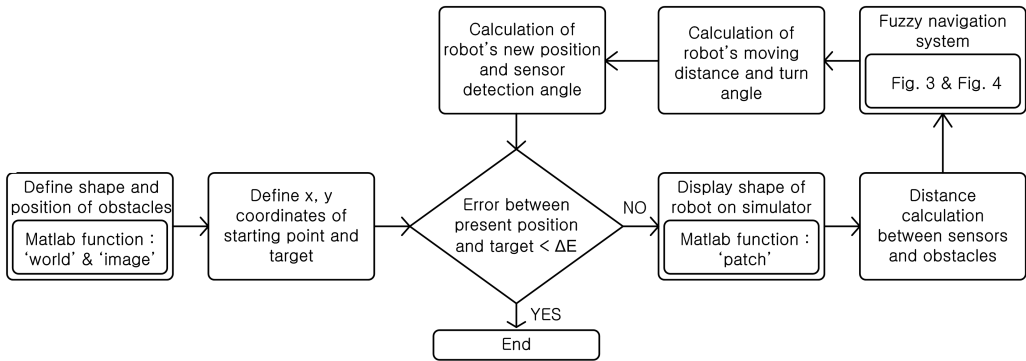


Fig. 7. Flowchart of data process in robot simulator.

2.6 모바일 로봇 시뮬레이터의 설계

본 연구에서는 모바일 로봇의 주행 알고리즘의 성능을 검증하기 위해 모바일 로봇의 움직임을 보여주는 Matlab 기반 시뮬레이터를 제작하여 제안된 알고리즘의 성능을 검증하는데 사용하였다.

로봇 시뮬레이터에서 로봇의 주행 환경을 구현하기 위해 Matlab에 내장되어있는 함수 'world'를 사용하여 행렬 형식으로 주행 환경의 크기를 정의하였다. 여기서 행렬 내부의 n×n 개의 값들은 각각의 색상을 나타내는 값들이며, 예를 들어 Matlab에서는 '1'은 흰색을 '0.5'는 회색을 나타낸다. 행렬로 정의된 주행 환경을 시각적으로 나타내기 위하여 행렬의 요소 값들이 가지는 색상을 그림으로 보여주는 함수인 'image'를 사용하였다. 시뮬레이션 환경은 앞서 설명한 두 가지의 함수를 사용하여 흰색으로 표현된 일반 주행 지형과 회색으로 표현되는 장애물들을 가지는 지형을 그림으로 나타내게 된다. 로봇의 형상은 시뮬레이션 환경에서 사각형으로 그려지며 가상 주행 환경속에서 보일 수 있도록 주행 방향의 실시간 변화에 따른 로봇의 좌표 변화를 계산하고 함수 'patch'를 사용하여 시뮬레이션 환경에서 나타나게 된다. 로봇의 주행 경로는 양쪽 바퀴의 회전 속력으로부터 계산된 로봇의 이동 거리와 진행 방향에 의해서 겹쳐서 생성되기 때문에 경로의 색깔이 짙을수록 로봇의 속도가 느림을 의미한다. 로봇에 탑재된 거리센서의 기능을 시뮬레이션 환경에서 구현하기 위해 로봇의 사각 형상에서 Fig. 2(a)와 같이 센서 각각의 설치 위치와 감지 각도를 정하고 감지 범위 내에 장애물이 있을 경우 장애물과 로봇과의 직선거리를 실시간으로 계산하여 2.1장에서 설명한 입력 신호들(d1~d8)로

사용하였다. Fig. 7은 시뮬레이터의 데이터 처리과정을 보여준다.

Fig. 8은 Fig. 5(a)의 입력 멤버 함수를 사용한 입력 센서 값들의 퍼지화를 위해 Matlab 기반 시뮬레이터 상에서 입력 센서 값들을 처리하는 과정을 보여준다. Fig. 7은 왼쪽 센서의 거리 데이터를 처리하는 과정을 나타내며, 전면과 오른쪽 거리센서 및 Fig. 5(b)의 멤버 함수를 사용한 GPS 센서 데이터의 처리과정도 이와 같은 방법으로 이루어 졌다. 퍼지화 과정에서 발생한 12개의 출력 변수들(L, F, R\_Close, L, F, R\_Med, L, F, R\_Far, L, S, R\_pos)은 퍼지화된 값들을 가지게 된다.

Fig. 9은 퍼지화된 센서 데이터를 사용한 규칙평가 및 비퍼지화 과정을 보여준다. 비퍼지화의 과정은 두 바퀴의 속도를 제어하는 신호를 생성하기 위해 각각 두 부분으로 나누어져 있으며, Fig. 9에서는 Rule Group A의 왼쪽 바퀴 제어를 위한 규칙평가 및 비퍼지화 과정을 보여준다. 최종적으로 비퍼지화

```

    if(L_distance < 10)
        L_Close = 1;
    elseif(L_distance >= 10 & L_distance < 20)
        L_Close = (-1/10)*L_distance + 2;
        L_Med = (1/10)*L_distance - 1;
    elseif(L_distance >=20 & L_distance < 40)
        L_Med = (-1/20)*L_distance + 2;
        L_Far = (1/20)*L_distance - 1;
    elseif(L_distance >=40)
        L_Far = 1;
    end
  
```

Fig. 8 Fuzzification procedure of sensor data.



```

function[LVel] = Rule_Eval_Defuzz(L_Close,L_Med,L_Far,F_Close,F_Med,
F_Far, ..., R_pos, Center_Slow, Center_Med, Center_Fast)

      81 Rules                                Center values in Fig.6

num_L = small(L_far,F_far,R_far,S_pos) * Center_Fast +
+ small(L_far,F_far,R_far,R_pos) * Center_Slow +
+ small(L_far,F_far,R_far,L_pos) * Center_Fast + ... ..
+ small(L_clo,F_clo,R_med,L_pos) * Center_Fast;

den_L = small(L_far,F_far,R_far,S_pos) +
+ small(L_far,F_far,R_far,R_pos) +
+ small(L_far,F_far,R_far,L_pos) + ... ..
+ small(L_clo,F_clo,R_med,L_pos);

LVel = num_L/den_L;
    
```

Fig. 9. Rule Evaluation and Defuzzification procedure.

과정에서 생성된 출력 값들(LVel, RVel)은 두 바퀴의 속도를 제어하는 신호가 된다.

Fig. 10은 퍼지 주행시스템의 출력으로부터 로봇의 이동 거리 및 회전 각도를 계산하여 새로운 위치 좌표를 생성하는 과정을 보여준다.

퍼지 주행 시스템의 출력으로부터 바퀴 중심의 선속도( $v_L$ ,  $v_R$ )는 식 (4)와 식 (5)로 정의된다. 여기서  $K$ 는 퍼지 시스템의 출력과 실제 바퀴의 회전속도( $\omega_L$ ,  $\omega_R$ ) 사이의 결합 상수로써 단위는 rad/s로 정의되며  $r$ 은 바퀴의 반지름을 나타낸다.

$$v_L = LVel \cdot K \cdot r = \omega_L \cdot r \tag{4}$$

$$v_R = RVel \cdot K \cdot r = \omega_R \cdot r \tag{5}$$

최종적으로 로봇의 회전 각도와 새로운 위치 좌표

는 식 (6)와 식(7)로부터 계산되며  $T$ 는 샘플링 간격을 나타낸다.

$$\theta(k+1) = \theta(k) + \left( \frac{v_R - v_L}{d} \right) T \tag{6}$$

$$\{x(k+1), y(k+1)\} = \{x(k), y(k)\} + \left( \frac{v_R + v_L}{2} \right) T \{\cos(k+1), \sin(k+1)\} \tag{7}$$

### 3. 결과 및 고찰

#### 3.1 단순 지형에서의 주행성능 평가

본 논문에서는 제안된 퍼지 주행 알고리즘의 성능을 Matlab 환경에서 개발된 로봇 주행 시뮬레이터를 사용하여 검증하였다. 이 장에서는 퍼지 알고리즘을 탑재한 로봇이 다양한 모양의 장애물들을 피하면서 최종 목표지점에 도달하기까지의 다양한 주행 경로를 보여준다.

Fig. 11의 결과는 간힘 현상이 발생하지 않는 장애물들로 구성된 지형에서 로봇의 주행 성능을 보여준다. Fig. 11(a)의 경우는 로봇이 두 개의 장애물들 너머에 있는 목표지점에 도달하기까지의 주행 경로이다. 시뮬레이션 결과로부터 로봇이 장애물들과 충돌 없이 목표지점에 성공적으로 도달함을 볼 수 있다. Fig. 11(b)의 경우는 장애물이 간힘 현상을 발생시킬 수 있는 형상임에도 불구하고 목표지점이 장애물 내부에 위치하고 있기 때문에 로봇이 어려움 없이 목표지점에 도달함을 볼 수 있다.

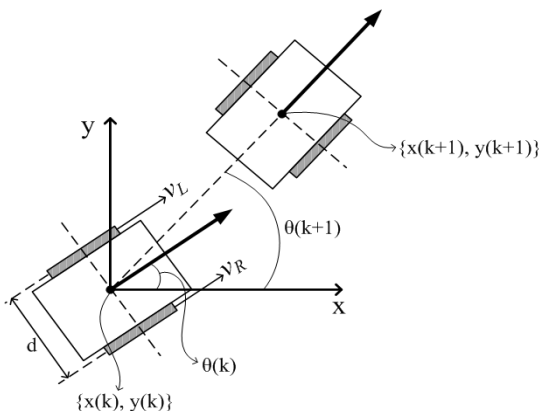
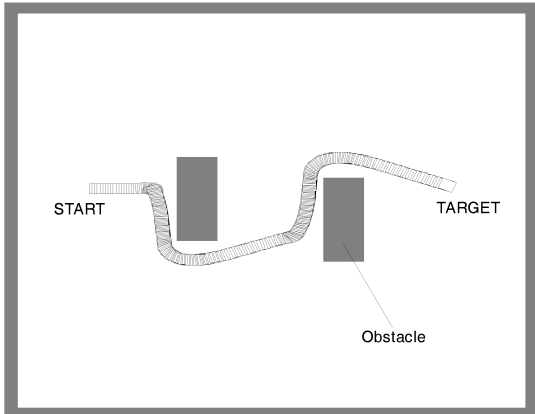
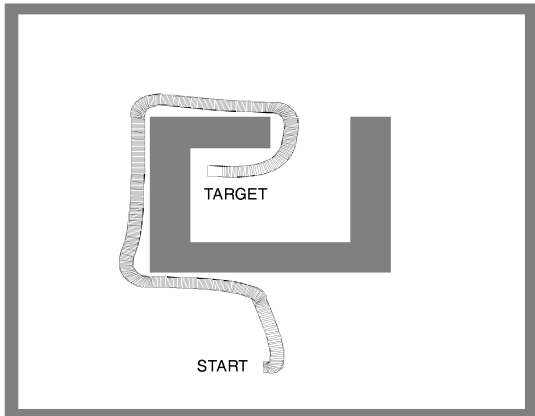


Fig. 10. Parameters in the simulation of robot motion.



(a)

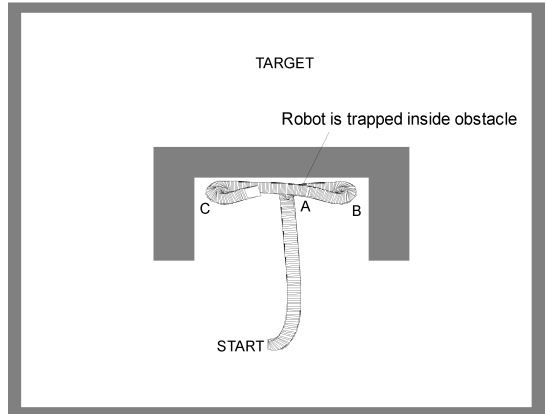


(b)

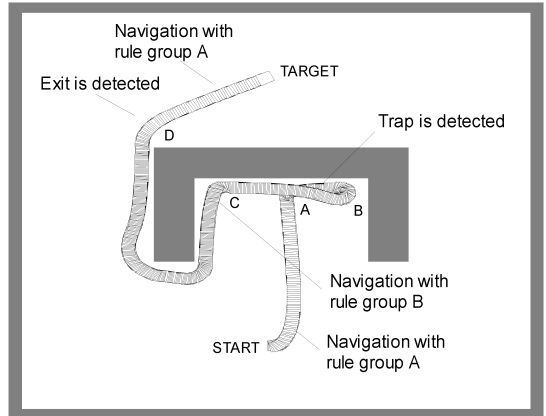
Fig. 11. Robot navigation in various environment.

### 3.2 장애물 탈출 성능 평가

Fig. 12에서는 로봇의 장애물 탈출 성능 평가를 위한 시뮬레이션 결과를 보여준다. Fig. 12(a)에서는 Rule group A로 구성된 주행알고리즘을 탑재한 로봇의 주행 경로를 보여주며, Fig. 12(b)에서는 Rule group A와 B로 구성된 주행알고리즘을 탑재한 로봇의 주행 경로를 보여준다. Fig. 12(a)에서 로봇이 장애물 너머에 있는 목표지점을 향해 이동하는 과정에서 A지점에 도달하게 되면 장애물 회피 규칙의 영향으로 B지점으로 이동하게 된다. B지점에서는 장애물 회피 규칙과 목표 지점 추종을 위한 규칙의 복합적인 영향으로 A지점을 지나 C지점으로 움직이게 된다. C 지점에서는 다시 장애물 회피 규칙과 목표 지점 추종을 위한 규칙의 영향으로 A지점을 지나 B 지점으로 움직이게 되고 반복적인 행위 규칙들의 영



(a)

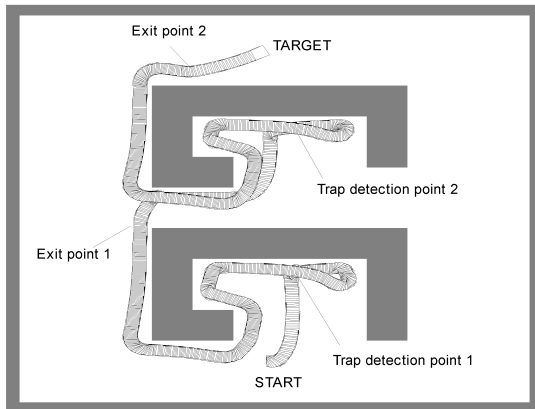


(b)

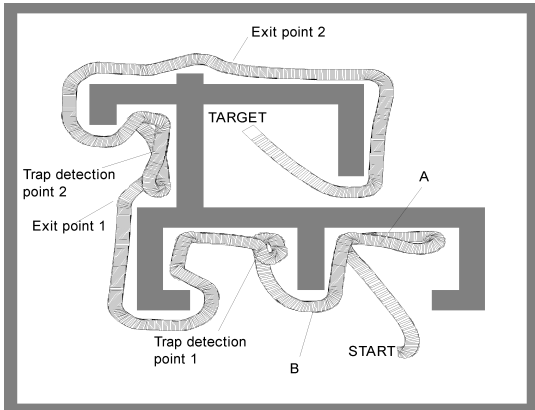
Fig. 12. Escaping ability of robot from obstacles with proposed fuzzy navigation algorithm: (a) navigation with rule group A, (b) navigation with both rule group A and B.

향으로 장애물을 벗어나지 못하고 내부에 갇히고 주행에 실패하게 됨을 시뮬레이션 결과로부터 볼 수 있다. 반면에 Fig. 12(b)에서는 로봇이 B지점에서 장애물 회피 규칙과 목표 지점 추종을 위한 규칙의 영향으로 회전하여 A지점으로 되돌아오는 과정에서 장애물속에 갇혔음을 감지하게 되고 Rule group B를 사용하여 장애물의 내부 가장자리를 따라 움직이게 된다. D 지점에서는 로봇과 목표지점과의 직선거리가 최초 갇힘 감지지점에서의 로봇과 목표지점과의 직선거리보다 짧음이 감지되기 때문에 다시 Rule group A를 사용하여 목표지점에 도달하게 된다.

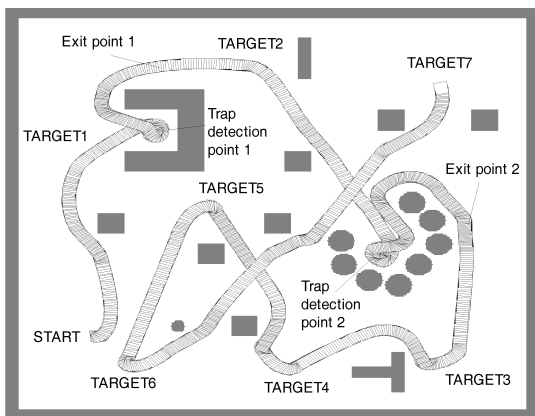
Fig. 13에서는 갇힘 현상을 야기할 수 있는 형태의 장애물들로 구성된 복잡한 환경 속에서 주행 알고리



(a)



(b)



(c)

Fig. 13. Robot motion in complex environment.

즘의 성능 평가를 위한 시뮬레이션 결과를 보여준다. Fig. 13(a)는 간힘 현상을 야기하는 이중의 장애물을 통과하여 목표지점에 성공적으로 도달하는 로봇의 주행 경로를 보여준다. 앞서 설명한 바와 같이 로봇

은 Fig. 13(a)의 주행 경로 상에 표시된 간힘 감지 지점들(Trap detection point 1 and 2)에 도달한 이후로 Rule group B를 사용하여 탈출 지점들(Exit 1 and 2)까지 이동하게 된다. 탈출 지점들이 감지되면 그 시점부터는 다시 Rule group A를 사용하여 목표지점으로 향하게 된다.

Fig. 13(b)에서는 간힘 현상을 야기하는 3개의 장애물들로 구성된 복잡한 주행환경 속에서 제안된 알고리즘의 성능 평가 결과를 보여준다. 첫 번째 장애물 속에서 A 지점을 반복해서 통과하였지만 첫 번째 통과 후 같은 지점으로 되돌아오기까지 걸리는 시간이 프로그램 되어있는  $\Delta t$ 의 범위를 벗어나기 때문에 동일 지점을 통과하는 것을 감지하지 못하고 계속해서 Rule group A를 사용하여 이동하게 된다. B 지점에서는 최종 목표지점의 위치가 그림에서 좌측 상단에 위치하기 때문에 목표 지점을 향해 Rule group A를 사용하여 두 번째 장애물 내부로 이동하게 된다. 두 번째 장애물의 내부에서는 Trap detection point 1 지점을 통과한 이후로 Rule group B를 사용하여 장애물의 가장자리를 따라 이동하게 되며 목표지점과의 직선거리가 간힘 감지 지점에서 목표지점과의 직선거리보다 짧아지는 지점(Exit point 1)에서 다시 Rule group A로 전환하게 된다. 세 번째 장애물의 내부에서도 간힘이 감지된 지점(Trap detection point 2)에서부터 Exit point 2 지점까지 Rule group B가 작동하게 되고 이후로는 Rule group A를 사용하여 최종 목표지점에 도달하게 된다. 시뮬레이션 결과로부터 본 연구에서 제안하는 퍼지 기반의 주행 알고리즘을 사용하는 로봇은 장애물 속에서 탈출하는 능력이 향상되었으며 결과적으로 주행 실패의 확률이 낮아졌음을 알 수 있다. 그러나 로봇의 이동 경로를 살펴보면 거리상 최적의 경로를 선택하지 못하는 단점을 보인다.

Fig. 13(c)는 간힘 현상을 야기하는 장애물들과 일반적인 형상의 장애물들이 산재되어있는 지형에서 로봇의 주행 성능을 보여주는 시뮬레이션의 결과이다. 로봇이 간힘 현상을 발생시키는 개별 장애물들을 성공적으로 탈출하여 여섯 개의 목표지점들을 지나 최종 목표지점에 도달함을 결과로부터 볼 수 있다. 일반적인 형상의 장애물들이 산재되어 있는 구간에서는 Rule group A의 장애물 회피 규칙과 목표 지점 추종을 위한 규칙의 작용에 의해서 장애물들과 충돌

을 피하는 범위 내에서 상대적으로 최적의 이동 경로를 따라 움직임을 볼 수 있다.

로봇이 최적의 이동 경로를 따라 움직이기 위해서 이동 방향을 계산하는 알고리즘들에 대한 연구가 많이 진행되어 왔다[11-20]. 이들 알고리즘 모두가 지역적인 정보를 가지는 맵을 활용하는 방법으로 지형에 대한 사전 정보를 바탕으로 최적 이동 경로를 계산하는 복잡한 과정이 필요하다. 로봇의 주행 알고리즘 개발 과정에서도 알고리즘의 복잡성과 주행상의 효율성 사이의 trade off 문제에 부딪치게 된다. 본 연구에서는 로봇이 지리에 대한 사전 정보 없이 단순한 거리센서와 GPS 센서만을 가지고 장애물에서 탈출하고 목표지점에 도달할 수 있는 방법을 제시하고 있으며, 장애물 탈출 성능이 향상되었음이 결과를 통하여 검증되었다.

#### 4. 결 론

로봇이 사전 정보 없이 목표지점을 찾아 항해하는 동안 다양한 형태의 장애물들을 만나게 된다. 이러한 다양한 주행상의 변수들을 사전에 예측하여 주행 프로그램으로 구현하는 것은 매우 어려운 문제이다. 이러한 주행 알고리즘의 복잡성을 해결하기에 적합한 것이 퍼지 이론이며, 이를 활용하여 변수들 사이에 미묘한 차이의 표현과 로봇에게 다양한 주행 기법을 제공하도록 프로그램 되어있는 행위 유닛들 간에 자연스러운 결합이 가능해졌다.

본 논문에서는 무인 모바일 로봇의 주행 성능 향상을 위해 퍼지화(Fuzzification), 규칙평가(Rule evaluation), 비퍼지화(Defuzzification) 단계로 구성된 퍼지 주행 알고리즘을 개발하고 그 성능을 시뮬레이션 환경에서 검증하였다. 특히 로봇이 주행 중에 U자 형상의 장애물 속에 갇히게 되는 기존 퍼지 주행 알고리즘의 단점을 극복하기 위해 두 가지 규칙 그룹이 복합된 형태의 규칙평가 단계를 가지는 퍼지 주행 알고리즘을 설계하여 기존 퍼지 주행알고리즘과 그 성능을 비교 분석하였다. 다양한 시뮬레이션 결과로부터 장애물 속에 갇혀 주행에 실패하게 되는 기존 퍼지 주행 알고리즘과는 달리 제안된 알고리즘을 사용할 경우 성공적으로 장애물을 벗어나 최종 목표지점에 로봇이 도달함을 검증할 수 있었다. 그러나 본 연구를 통하여 제안된 알고리즘으로도 미결된 과제가 여전히 남아 있다. 첫째로 로봇이 장애물 속에 갇

혔음을 판단하는 알고리즘이 개선되어야만 한다. 본 연구에서는 일정 간격( $\Delta t$ ) 만큼 떨어진 데이터를 지속적으로 비교하여 유사성으로 간헐을 판단하였는데, 동일 지점에 되돌아 올 때까지의 시간이 사전 정의된  $\Delta t$  보다 길거나 짧을 경우 간헐을 인식하지 못하게 된다. 둘째로, 간헐이 감지된 지점에서 Rule group B가 작동하면서 로봇이 지나온 경로로 되돌아가는 경우가 발생할 수 있다. 이 경우 새로운 국소 목표지점을 생성하여 로봇의 경로를 일시적으로 바꾸는 방법을 적용할 수 있지만 지형에 대한 사전 정보가 없을 경우 잘못된 국소 목표지점의 지정으로 주행 경로가 더욱 복잡해지거나 국소 목표지점이 다른 장애물상에 놓이게 되는 경우 주행 실패를 야기한다. 따라서 지형에 대한 전반적인 정보를 활용하거나 비전 센서 등 다양한 성능의 센서들을 활용하여 지형에 대한 패턴 인식을 통한 판별력을 높이는 기술의 적용이 필요하다. 이러한 제한적인 성능에도 불구하고 본 논문은 거리센서와 GPS 센서의 단순 기능만을 사용하여 로봇이 장애물 속에 갇히는 문제를 해결하고자 노력하였으며, 논문에서 제안된 알고리즘의 사용으로 로봇의 장애물 탈출 성능이 향상되었으며 로봇이 다양한 장애물들을 자연스럽게 피하고 목표지점에 도달함을 검증하였다.

향후 지속적인 연구를 통하여 주행 알고리즘의 성능의 개선 및 복잡한 장애물 속에서의 최적 경로를 찾아 주행의 효율성을 향상시킬 수 있는 알고리즘 개발 연구를 수행하고자 한다.

#### REFERENCE

- [1] F. Kabanza and K.B. Lamine, "Specifying Failure and Progress Conditions in a Behavior-Based Robot Programming System," *Proceeding of the 4th International Cognitive Robotics Workshop*, pp. 11-16, 2004.
- [2] A. Saffiotti, E.H. Ruspini, and Konolige, *Chapter 5 of the International Handbook of Fuzzy Sets*, Kluwer Academic Publishers, Netherlands, 1999.
- [3] W. Li, "Fuzzy Logic-Based 'Perception-Action' Behavior Control of Mobile Robot in Uncertain Environments," *Proceeding of IEEE*

- International Conference on Fuzzy Systems*, Vol. 3, pp. 1626-1631, 1994.
- [4] K. Song and J. Tai, "Fuzzy Navigation of a Mobile Robot," *Proceeding of IEEE International Conference on Intelligent Robots and Systems*, Vol. 1, pp. 621-627, 1992.
- [5] A. Ramirez-Serrano and M. Boumedine, "Ultrasonic Sensing and Fuzzy Logic Control for Navigation in Unknown Static Environments," *Proceeding of Eurobot*, pp. 54-59, 1996.
- [6] N. Zhang, D. Beetner, D.C. Wunsch II, B. Hemmelman, and A. Hasan, "An Embedded Real-time Neuro-fuzzy Controller for Mobile Robot Navigation," *Proceeding of IEEE International Conference on Fuzzy Systems*, pp. 319-323, 2005.
- [7] V. Sezer and M. Gokasan, "A Novel Obstacle Avoidance Algorithm: Follow the Gap Method," *Robotics and Autonomous Systems*, Vol. 60, Issue 9, pp. 1123-1134, 2012.
- [8] A. Yufka and O. Parlaktuna, "Performance Comparison of Bug Algorithms for Mobile Robots," *Proceeding of the 5<sup>th</sup> International Advanced Technology Symposium*, pp. 61-65, 2009.
- [9] J. Oroko and B. Ikua, "Obstacle Avoidance and Path Planning Schemes for Autonomous Navigation of a Mobile Robot: A Review," *Proceeding of Mechanical Engineering Conference on Sustainable Research and Innovation*, pp. 314-318, 2012.
- [10] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An Effect Improved Artificial Potential Field Based Regression Search Method for Robot Path Planning," *Proceeding of International Conference on Mechatronics and Automation*, pp. 1227-1232, 2012.
- [11] Y. Zhu, T. Zhang, J. Song, and X. Li, "A New Hybrid Navigation Algorithm for Mobile Robots in Environments with Incomplete Knowledge," *Knowledge-Based Systems*, Vol. 27, Issue 1, pp. 302-313, 2012.
- [12] A. Sgorbissa and R. Zaccaria, "Planning and Obstacle Avoidance in Mobile Robotics," *Robotics and Autonomous Systems*, Vol. 60, Issue 4, pp. 628-638, 2012.
- [13] K.H. Chen and W.H. Tsai, "Vision-Based Obstacle Detection and Avoidance for Autonomous Land Vehicle Navigation in Outdoor Roads," *Automation in Construction*, Vol. 10, Issue 1, pp. 1-25, 2000.
- [14] S.G. Tzafestas and K.C. Zikidis, "A 3-level Neuro-fuzzy Autonomous Robot Navigation System," *Proceeding of IEEE International Conference on Fuzzy Systems*, pp. 1626-1631, 1994.
- [15] H. Surmann, J. Huser, and L. Peters, "A Fuzzy System for Indoor Mobile Robot Navigation," *Proceeding of the Fourth IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 83-86, 1995.
- [16] S. Thongchai, S. Suksakulchai, D.M. Wilkes, and N. Sarkar, "Sonar Behavior-Based Fuzzy Control for a Mobile Robot," *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 3532-3537, 2000.
- [17] M. Benreguieg, P. Hoppenot, H. Maaref, E. Colle, and C. Barret, "Control of a Medical Aid Mobile Robot based on a Fuzzy Navigation," *Proceeding of IEEE Symposium on Robotics and Cybernetics*, pp. 388-393, 1996.
- [18] P. Rusu, E.M. Petriu, T.E. Whalen, A. Cornell, and H.J.W. Spoelder, "Behavior-Based Neuro-Fuzzy Controller for Mobile Robot Navigation," *IEEE Transaction on Instrumentation and Measurement*, Vol. 52, No. 4, pp. 1335-1340, 2003.
- [19] A.O. Djekoune, K. Achour, and R. Toumi, "A Sensor Based Navigation Algorithm for a Mobile Robot using the DVFF Approach," *International Journal of Advanced Robotic Systems*, Vol. 6, No. 2, pp. 97-108, 2009.

- [20] E. Lim and K. Kim, "A Study on the Recognition of Car Plate using an Enhanced Fuzzy ART Algorithm," *Journal of Korea Multimedia Society*, Vol. 3, No. 5, pp. 433-444, 2000.



박 기 원

2011년 미국 South Dakota School of Mines & Technology 박사

현재 : 영산대학교 그린자동차학과 교수

관심분야 : 이온 전도성 폴리머, 수소 연료전지, 전기자동차 동력 시스템