

# 이동 로봇의 장애물회피를 위한 퍼지제어기와 실시간 제어시스템 적용을 위한 저(低)복잡도 검색테이블 공유기법

## A Fuzzy Controller for Obstacle Avoidance Robots and Lower Complexity Lookup-Table Sharing Method Applicable to Real-time Control Systems

김진욱<sup>1</sup>, 김윤구<sup>1</sup>, 안진웅<sup>1,✉</sup>  
Jinwook Kim<sup>1</sup>, Yoon-Gu Kim<sup>1</sup> and Jinung An<sup>1,✉</sup>

<sup>1</sup> 대구경북과학기술원 실용로봇연구소 (Pragmatic Applied Robot Institute, Daegu Gyeongbuk Institute of Science & Technology)

✉ Corresponding author: robot@dgist.ac.kr, Tel: 053-430-8512

Manuscript received: 2009.9.4 / Revised: 2009.11.25 / Accepted: 2009.11.25

*Lookup-Table (LUT) based fuzzy controller for obstacle avoidance enhances operations faster in multiple obstacles environment. An LUT based fuzzy controller with Positive/Negative (P/N) fuzzy rule base consisting of 18 rules was introduced in our paper<sup>1</sup> and this paper shows a 50-rule P/N fuzzy controller for enhancing performance in obstacle avoidance. As a rule, the more rules are necessary, the more buffers are required. This paper suggests LUT sharing method in order to reduce LUT buffer size without significant degradation of performance. The LUT sharing method makes buffer size independent of the whole fuzzy system's complexity. Simulation using MSRDS(MicroSoft Robotics Developer Studio) evaluates the proposed method, and in order to investigate its performance, experiments are carried out to Pioneer P3-DX in the LabVIEW environment. The simulation and experiments show little difference between the fully valued LUT-based method and the LUT sharing method in operation times. On the other hand, LUT sharing method reduced its buffer size by about 95% of full valued LUT-based design.*

Key Words: Autonomous Navigation (자율주행), Obstacle Avoidance (장애물 회피), Fuzzy Controller (퍼지 제어기), Mobile Robot (이동 로봇), Lookup Table (검색테이블)

### 기호설명

- c = center of the distribution
- $\sigma$  = variation of the function
- $\omega$  = weights of output fuzzy sets
- $\mu$  = membership degree of each fuzzy set
- y = output value of the 50-rule P/N fuzzy controller

### 1. Introduction

로봇이 자율적으로 목적지로 이동하게 하기 위하여 모바일 로봇의 자율주행 기술에 많은 연구가

진행되고 있다. 이를 위해 자율주행 과정에서는 환경지도를 작성하는 것이 필요한데, 이는 로봇이 주행 중에 센서로부터 장애물에 대한 정보를 얻고 이들의 위치를 계산하여 지도의 해당하는 위치에 장애물에 대한 정보를 기록하는 과정이다. 로봇의 위치에서 목적지로의 주행경로는 경로생성 당시까지 지도를 통해 알려진 정보를 이용하여 생성된다. 만약 주행 중에 지도에 알려진 장애물 이외의 장애물이 나타나지 않으면 생성된 경로에 의해 로봇을 목표점으로 안정적으로 이동할 수 있으므로 장애물회피는 필요하지 않다. 하지만 로봇의 주행경로 상에 알려지지 않은 장애물이 센서에 감지가

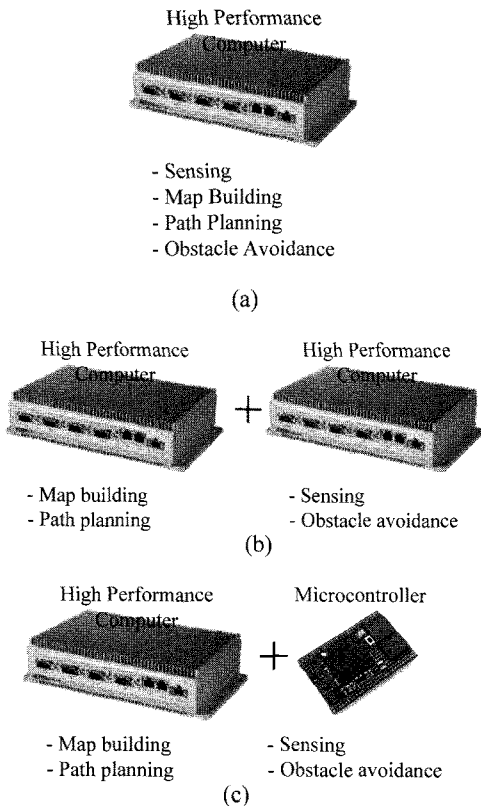


Fig. 1 Example of mobile robot controller composition

되면 이 장애물을 회피하면서 주어진 경로를 주행하므로 장애물 회피 기술이 필요하다. 예측하지 못한 장애물을 실시간으로 회피하고 목표점으로 안정적인 주행을 하기 위해서는 로봇이 장애물을 감지함과 동시에 주행각도와 주행속도를 빠르게 결정해야 한다. 또한 장애물 회피 과정은 로봇의 주행에 있어서 충돌을 막기 위해서 매우 중요하므로 짧은 주기의 반복적인 제어가 필요하다. 이를 위하여 로봇 제어기 중 장애물 회피 제어기는 컴퓨팅 과정에서 다른 장치 및 연산에서 발생하는 인터럽트(interrupt) 등의 요소가 발생하지 않도록 독립적인 컴퓨팅 시스템을 사용하는 것이 바람직하다. 하지만 로봇의 장애물 회피를 위하여 높은 컴퓨팅 성능을 가지는 컴퓨터 시스템을 추가적으로 사용하는 것은 로봇의 시스템 비용 측면이나 전력 소모 등을 고려하였을 때 적절하지 못한 방법이다. Fig. 1 에 모바일로봇 제어기의 구성이 예시되어 있는데 (a)는 하나의 컴퓨터 시스템이 모든 제어를 담당하므로 인터럽트나 시스템 예외(exception)가 발생했을 때 안정적인 장애물 회피가

어려워 질 수 있으며, (b)는 두 대의 고성능컴퓨터를 사용함으로써 시스템 구성에서의 비용과 전력 소모가 증가되는 단점이 있다. 따라서 장애물 회피 제어기의 독립적인 컴퓨팅 시스템을 구성하기 위해 (c)에서와 같이 마이크로컨트롤러 등의 임베디드 프로세서를 사용하는 것이 적절한 것으로 보인다. 이를 위하여 본 논문에서는 장애물회피 퍼지제어기를 설계하고 이것의 연산 처리의 성능을 향상시켜 빠른 장애물 회피가 가능하면서도 시스템 자원의 사용을 최소화하여 임베디드 프로세서에 적용하기 적합하도록 한 제어기 구현 방법을 제안한다.

장애물 회피를 위해서 potential field algorithms,<sup>2</sup> vector field histogram algorithms,<sup>3</sup> fuzzy controller algorithms<sup>4</sup> 등의 다양한 알고리즘이 연구되고 있다. 기본적인 퍼지 제어기 알고리즘은 Wang,<sup>5</sup> Palm,<sup>6</sup> Passino,<sup>7</sup> Aguirre<sup>8</sup> 등의 논문에 소개되어 있다. 장애물 회피에서 이러한 퍼지 제어기 방법은 단순하고 직관적인 방법을 사용함으로써 실시간 시스템의 구성에 유리하다. Mbede<sup>9</sup> 는 신경-퍼지 모션 제어기(neuro-fuzzy motion controller)를 제안하였고, Lilly<sup>10</sup> 는 P/N 퍼지 장애물 회피 제어기(P/N fuzzy obstacle avoidance controller)를 제안하였다. Lilly 의 연구는 부정 퍼지 법칙(negative fuzzy rules)을 사용하여 좀 더 직관적이고 법칙의 과잉(redundancy)을 줄인 방법을 제시하였다. Kiendl<sup>11</sup> 과 Branson<sup>12</sup> 또한 부정 퍼지 법칙의 개념을 사용한 퍼지 시스템을 제안하였다. 이러한 연구들 중에서 본 논문은 Lilly<sup>10</sup> 의 P/N 퍼지 장애물 회피 제어기 이론을 기반으로 진행된 연구에 관한 것이다. 본 논문의 연구에서 Lilly<sup>10</sup> 의 이론을 실제적으로 적용하기 위하여 개선된 알고리즘을 제안하였고, 그 방법을 로보틱스 시뮬레이터인 MSRDS(Microsoft Robotics Development Studio) 2008<sup>13</sup> 와 실제적인 실험으로 장애물 회피 주행을 검증하였다.

2 장에서 P/N 퍼지 장애물 회피 제어기 이론을 소개하고 50-Rule P/N 퍼지 제어기를 제안한다. 3 장에서 전역 검색 테이블 기반 구조를 소개하고, 4 장에서 검색 테이블 공유 기법을 제안한다. 제안한 방법으로서의 시뮬레이션 결과의 비교는 5 장에 보여주며, 6 장에 본 논문에 대한 결론을 기술한다.

## 2. P/N rule fuzzy controller

기존의 퍼지 법칙은 특정한 출력을 유도하는 역할을 한다. 하지만 P/N 퍼지 시스템 이론에서는

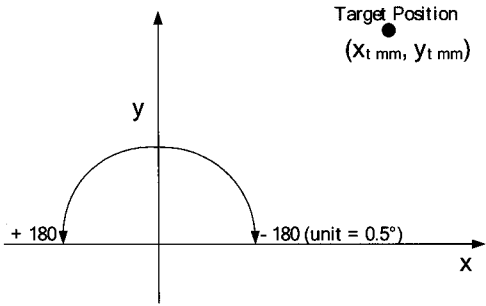


Fig. 2 Coordinates of the controller

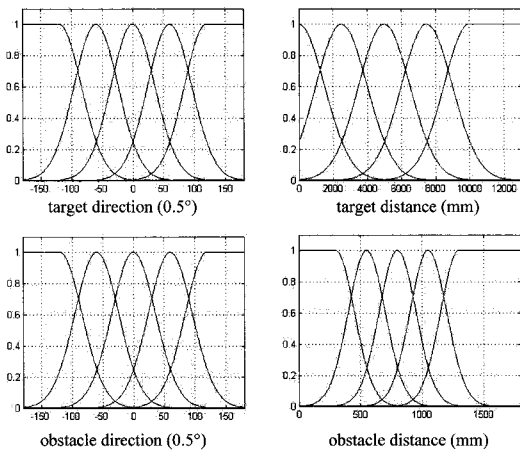


Fig. 3 Membership functions of input fuzzy set

부정 법칙의 개념이 적용되어 특정한 퍼지 출력을 억제하는 역할을 하는 퍼지 법칙이 추가된다. 따라서 P/N 퍼지 시스템에서는 두 가지 서로 다른 역할의 법칙이 존재하며, 긍정 법칙은 적절한 퍼지 출력을 유도하고 부정 법칙은 부적절한 퍼지 출력을 억제한다. P/N 퍼지 시스템을 장애물 회피에 적용하면 긍정 법칙이 로봇을 목표 지점을 향하도록 하는 역할을 하도록 하고, 부정 법칙이 로봇이 장애물을 향하지 않도록 하는 역할을 부여할 수 있다. 이러한 제어기에서 각 퍼지 법칙의 영향력은 장애물이 많지 않은 지역에서는 긍정 법칙이 지배적이고, 장애물 밀집 지역에서는 부정 법칙이 지배적으로 작용한다.

**2.1 Design of 50-Rule P/N Fuzzy Controller**

본 논문의 퍼지 제어기는 두 가지 종류의 입력을 기반으로 설계되었다. 그 중 하나는 로봇의 현재의 위치를 기준으로 한 목표지점의 상대적인 좌표이고, 나머지 하나는 레이저 센서 입력이다. 레

Table 1 Characteristics of fuzzy membership functions

		Target		Obstacle	
		Center (C)	Variation (σ)	Center (C)	Variation (σ)
Direction (0.5°)	HL	-120	36	-120	36
	SL	-60	36	-60	36
	S	0	36	0	36
	SR	60	36	60	36
	HR	120	36	120	36
Distance (mm)	Z	0	1500	300	150
	VN	2500	1500	550	150
	N	5000	1500	800	150
	F	7500	1500	1050	150
	VF	10000	1500	1300	150

이저 센서로 사용할 LRF(Laser Range Finder) 센서는 180°각도 범위에서 0.5° 각도 단위의 장애물 거리 측정치를 제공한다. 본 논문에서는 제어기 시스템을 단순화하기 위하여 0.5° 라는 단위의 크기를 임의적으로 두 배로 하여 단위를 1°로 조정하였다. 그리고 목표 지점까지의 거리는 밀리미터 (milli-meter) 단위를 사용하였는데, Fig. 2에 각도와 거리의 단위가 나타나 있다. 목표 지점의 좌표가 주행 지도에서 선택이 되면, 그 값은 지도상에서는 고정된 값이지만 로봇이 이동하면서 상대적인 좌표가 변화하게 되므로 현재 로봇의 위치를 기준으로 하여 목표 지점의 상대적인 좌표를 주행 중에 갱신해야 한다. 앞으로 설명될 50-Rule P/N 퍼지 제어기에서 25 개의 긍정 법칙은 목표지점의 상대적인 좌표와 관련이 되며, 25 개의 부정 법칙은 레이저 센서로부터 입력 받은 장애물의 위치 정보와 관련이 된다. 퍼지 제어기의 입력은 목표 지점의 방향과 거리, 그리고 장애물의 방향과 거리가 된다. 최종적인 제어기의 출력은 로봇의 진행 방향과 속도이다.

**2.2 Fuzzification**

퍼지 멤버함수는 Fig. 3에 나타나 있고 각 퍼지 함수는 수식 (1)의 가우시안 곡선을 따른다.

$$f(x) = e^{-\frac{(x-c)^2}{2\sigma^2}} \tag{1}$$

여기서 c 는 이 분포의 중심을 나타내며, σ 는 이 분포의 분산을 나타낸다. Table 1 은 Fig. 3에 나타

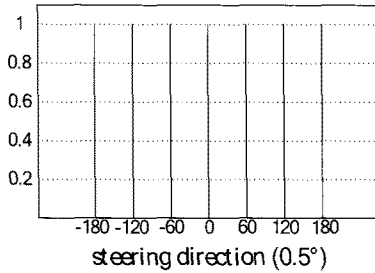


Fig. 4 Membership function of output fuzzy set

나 있는 각 멤버함수의  $c$  와  $\sigma$  를 나타낸다. 퍼지 집합 Hard Left(HL), Soft Left(SL), Straight(S), Soft Right(SR), and Hard Right(HR)은 방향과 관계되는 퍼지 집합인데 입력으로 들어오는 값 중 목표지점에 대한 방향과 장애물에 대한 방향, 두 값에 대해 공통적으로 사용된다. 만약 목표지점의 방향이 정확하게 정면을 가리킨다면 해당하는 퍼지 입력에 대한 S 의 멤버함수 값은 1 이고, HL 의 멤버함수 값은 0 이 된다. 또한 Zero(Z), Very Near(VN), Near(N), Far(Far), and Very Far(VF)은 입력 값 중 목표점까지의 거리와 장애물까지의 거리에 대해 사용된다. 만약 장애물지점의 거리 값이 2000mm 이상이라면 거리 입력에 대한 Z 의 멤버함수 값은 0 이고, VF 의 멤버함수 값은 1 이 된다. 출력 퍼지 집합 Hard Left(HL), Left(L), Soft Left(SL), Straight(S), Soft Right(SR), Right(R), Hard Right(HR)은 긍정 범칙에 관계된 것이며, 각 긍정범칙의 퍼지 집합에 대한 여집합의 성격을 가지는  $\overline{HL}$ ,  $\overline{L}$ ,  $\overline{SL}$ ,  $\overline{S}$ ,  $\overline{SR}$ ,  $\overline{R}$ ,  $\overline{HR}$  은 부정 범칙에 관계된 것이다. 출력 퍼지 집합은 Fig. 4 의 멤버함수에 의해 특징지어지는데, 만약 출력 퍼지 집합 R 에 대한 적합도가 다른 출력 퍼지 집합에 비해 비교적 크다면 로봇의 진행 방향이 60°로 결정이 되는 것이 비교적 적합하다는 의미를 가진다.

**2.3 Rule Base**

Table 2 와 Table 3 에 각각 긍정범칙과 부정범칙의 범칙베이스(rule-base)이 나타나있다. 긍정범칙의 범칙베이스는 아래의 범칙 Rule 1P ~ Rule 25P 으로 설명될 수 있다.

Rule 1P

If  $x_{TargetDirection}$  is HL and  $x_{TargetDistance}$  is Z then y is HL.

Rule 2P

If  $x_{TargetDirection}$  is HL and  $x_{TargetDistance}$  is VN then y is HL.

Rule 3P

If  $x_{TargetDirection}$  is HL and  $x_{TargetDistance}$  is N then y is L.

Rule 4P

If  $x_{TargetDirection}$  is HL and  $x_{TargetDistance}$  is F then y is L.

Rule 5P

If  $x_{TargetDirection}$  is HL and  $x_{TargetDistance}$  is VF then y is SL.

Rule 6P

If  $x_{TargetDirection}$  is SL and  $x_{TargetDistance}$  is Z then y is HL.

Rule 7P

If  $x_{TargetDirection}$  is SL and  $x_{TargetDistance}$  is VN then y is L.

Rule 8P

If  $x_{TargetDirection}$  is SL and  $x_{TargetDistance}$  is N then y is L.

Rule 9P

If  $x_{TargetDirection}$  is SL and  $x_{TargetDistance}$  is F then y is SL.

Rule 10P

If  $x_{TargetDirection}$  is SL and  $x_{TargetDistance}$  is VF then y is S.

Rule 11P

If  $x_{TargetDirection}$  is S and  $x_{TargetDistance}$  is Z then y is S.

Rule 12P

If  $x_{TargetDirection}$  is S and  $x_{TargetDistance}$  is VN then y is S.

Rule 13P

If  $x_{TargetDirection}$  is S and  $x_{TargetDistance}$  is N then y is S.

Rule 14P

If  $x_{TargetDirection}$  is S and  $x_{TargetDistance}$  is F then y is S.

Rule 15P

If  $x_{TargetDirection}$  is S and  $x_{TargetDistance}$  is VF then y is S.

Rule 16P

If  $x_{TargetDirection}$  is SR and  $x_{TargetDistance}$  is Z then y is HR.

Rule 17P

If  $x_{TargetDirection}$  is SR and  $x_{TargetDistance}$  is VN then y is R.

Rule 18P

If  $x_{TargetDirection}$  is SR and  $x_{TargetDistance}$  is N then y is R.

Rule 19P

If  $x_{TargetDirection}$  is SR and  $x_{TargetDistance}$  is F then y is SR.

Rule 20P

If  $x_{TargetDirection}$  is SR and  $x_{TargetDistance}$  is VF then y is S.

Rule 21P

If  $x_{TargetDirection}$  is HR and  $x_{TargetDistance}$  is Z then y is HR.

Rule 22P

If  $x_{TargetDirection}$  is HR and  $x_{TargetDistance}$  is VN then y is HR.

Rule 23P

If  $x_{TargetDirection}$  is HR and  $x_{TargetDistance}$  is N then y is R.

Rule 24P

If  $x_{TargetDirection}$  is HR and  $x_{TargetDistance}$  is F then y is R.

Rule 25P

If  $x_{TargetDirection}$  is HR and  $x_{TargetDistance}$  is VF then y is SR.

Table 2 Rule base of 25-rule positive fuzzy system (expert provided)

Steering Direction		Target Direction				
		HL	SL	S	SR	HR
Target Distance	Z	HL	HL	S	HR	HR
	VN	HL	L	S	R	HR
	N	L	L	S	R	R
	F	L	SL	S	SR	R
	VF	SL	S	S	S	SR

Table 3 Rule base of 25-rule negative fuzzy system (expert provided)

Steering Direction		Obstacle Direction				
		HL	SL	S	SR	HR
Obstacle Distance	Z	$\overline{HL}$	$\overline{HL}$	$\overline{S}$	$\overline{HR}$	$\overline{HR}$
	VN	$\overline{HL}$	$\overline{L}$	$\overline{S}$	$\overline{R}$	$\overline{HR}$
	N	$\overline{L}$	$\overline{L}$	$\overline{S}$	$\overline{R}$	$\overline{R}$
	F	$\overline{L}$	$\overline{SL}$	$\overline{S}$	$\overline{SR}$	$\overline{R}$
	VF	$\overline{SL}$	$\overline{S}$	$\overline{S}$	$\overline{S}$	$\overline{SR}$

위의 법칙들 중에서 Rule 1P 를 예들들어 설명 하면 목표점이 HL(Hard Left : 매우 왼쪽에 있음)이고 목표점까지의 거리가 Z(Zero : 매우 가까움)일때 로봇의 주행방향은 HL(Hard Left : 매우 왼쪽)으로 주행하라는 법칙을 정의한 것이 된다. 그리고 부정법칙의 법칙베이스는 다음의 법칙 Rule 1N ~ Rule 25N 으로 설명될 수 있다.

Rule 1N

If  $x_{ObstacleDirection}$  is HL and  $x_{ObstacleDistance}$  is Z then y is not HL.

Rule 2N

If  $x_{ObstacleDirection}$  is HL and  $x_{ObstacleDistance}$  is VN then y is not HL.

Rule 25N

If  $x_{ObstacleDirection}$  is HR and  $x_{ObstacleDistance}$  is VF then y is not SR.

각 퍼지 집합에 대한 적합도는 Table 4 와 같이 표기될 수 있다. 긍정법칙 Rule 1P 에 해당하는 적합도는 목표지점의 방향에 대한 적합도 중에 하나인  $\mu_{HL}$  과 거리에 대한 적합도 중에 하나인  $\mu_Z$  의 곱에 오프셋 값을 더해준 값이 된다. 입력 퍼지집합 HL, Z 에 대한 출력 퍼지집합은 HL 인데, 위의 긍정 법칙 Rule 1P 에 대한 적합도는 HL 방향으로

Table 4 Membership degrees of input fuzzy sets

Degrees of positive-rule fuzzy sets		Degree of negative-rule fuzzy sets	
HL	$\mu_{HL}$	HL	$\mu_{\overline{HL}}$
SL	$\mu_{SL}$	SL	$\mu_{\overline{SL}}$
S	$\mu_S$	S	$\mu_{\overline{S}}$
SR	$\mu_{SR}$	SR	$\mu_{\overline{SR}}$
HR	$\mu_{HR}$	HR	$\mu_{\overline{HR}}$
Z	$\mu_Z$	Z	$\mu_{\overline{Z}}$
VN	$\mu_{VN}$	VN	$\mu_{\overline{VN}}$
N	$\mu_N$	N	$\mu_{\overline{N}}$
F	$\mu_F$	F	$\mu_{\overline{F}}$
VF	$\mu_{VF}$	VF	$\mu_{\overline{VF}}$

향하는 적합도라는 것을 의미한다. 이 값에 HL 방향으로 향하지 못하도록 하는 부정법칙에 대한 적합도를 곱하면 아래의 (2)와 같은 식이 된다.

$$\mu_1 = (\text{offset} + \mu_{HL} \cdot \mu_Z) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{Z_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{Z_i}) \quad (2)$$

나머지  $\mu_2 \sim \mu_{25}$  에 대한 적합도는 위와 같은 방식으로 계산되며 다음의 수식 (3)과 같다.

$$\begin{aligned} \mu_2 &= (\text{offset} + \mu_{HL} \cdot \mu_{VN}) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{Z_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{Z_i}), \\ \mu_3 &= (\text{offset} + \mu_{HL} \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{N_i}), \\ \mu_4 &= (\text{offset} + \mu_{HL} \cdot \mu_F) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{N_i}), \\ \mu_5 &= (\text{offset} + \mu_{HL} \cdot \mu_{VF}) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{VF}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{F}_i}), \\ \mu_6 &= (\text{offset} + \mu_{SL} \cdot \mu_Z) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{Z_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{Z_i}), \\ \mu_7 &= (\text{offset} + \mu_{SL} \cdot \mu_{VN}) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{N_i}), \\ \mu_8 &= (\text{offset} + \mu_{SL} \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{N_i}), \\ \mu_9 &= (\text{offset} + \mu_{SL} \cdot \mu_F) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{HL}_i} \cdot \mu_{\overline{VF}_i})(1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{F}_i}), \\ \mu_{10} &= (\text{offset} + \mu_{SL} \cdot \mu_{VF}) \\ &\quad \times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VF}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{Z_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{VF}_i}), \\ \mu_{11} &= (\text{offset} + \mu_S \cdot \mu_Z) \\ &\quad \times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VF}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{Z_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{VF}_i}), \\ \mu_{12} &= (\text{offset} + \mu_S \cdot \mu_{VN}) \\ &\quad \times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{\overline{SL}_i} \cdot \mu_{\overline{VF}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{Z_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{VN}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{N_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{F}_i})(1 - \mu_{\overline{S}_i} \cdot \mu_{\overline{VF}_i}), \end{aligned}$$

$$\begin{aligned}
 \mu_{13} &= (\text{offset} + \mu_S \cdot \mu_N) \\
 &\times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SL_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{Z_i}) (1 - \mu_{SR_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{N_i}) (1 - \mu_{S_i} \cdot \mu_{F_i}) (1 - \mu_{S_i} \cdot \mu_{V_i}) \\
 \mu_{14} &= (\text{offset} + \mu_S \cdot \mu_F) \\
 &\times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SL_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{Z_i}) (1 - \mu_{SR_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{N_i}) (1 - \mu_{S_i} \cdot \mu_{F_i}) (1 - \mu_{S_i} \cdot \mu_{V_i}) \\
 \mu_{15} &= (\text{offset} + \mu_S \cdot \mu_F) \\
 &\times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SL_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{Z_i}) (1 - \mu_{SR_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{N_i}) (1 - \mu_{S_i} \cdot \mu_{F_i}) (1 - \mu_{S_i} \cdot \mu_{V_i}) \\
 \mu_{16} &= (\text{offset} + \mu_{SR} \cdot \mu_Z) \prod_{i=1}^{\#ofObstacles} (1 - \mu_{SR_i} \cdot \mu_{Z_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) \\
 \mu_{17} &= (\text{offset} + \mu_{SR} \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SR_i} \cdot \mu_{N_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) (1 - \mu_{HR_i} \cdot \mu_{F_i}) \\
 \mu_{18} &= (\text{offset} + \mu_{SR} \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SR_i} \cdot \mu_{N_i}) (1 - \mu_{SR_i} \cdot \mu_{V_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) (1 - \mu_{HR_i} \cdot \mu_{F_i}) \\
 \mu_{19} &= (\text{offset} + \mu_{SR} \cdot \mu_F) \prod_{i=1}^{\#ofObstacles} (1 - \mu_{SR_i} \cdot \mu_{F_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) \\
 \mu_{20} &= (\text{offset} + \mu_S \cdot \mu_F) \\
 &\times \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SL_i} \cdot \mu_{V_i}) (1 - \mu_{S_i} \cdot \mu_{Z_i}) (1 - \mu_{S_i} \cdot \mu_{N_i}) (1 - \mu_{S_i} \cdot \mu_{F_i}) (1 - \mu_{S_i} \cdot \mu_{V_i}) \\
 \mu_{21} &= (\text{offset} + \mu_{HR} \cdot \mu_Z) \prod_{i=1}^{\#ofObstacles} (1 - \mu_{SR_i} \cdot \mu_{Z_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) \\
 \mu_{22} &= (\text{offset} + \mu_{HR} \cdot \mu_N) \prod_{i=1}^{\#ofObstacles} (1 - \mu_{SR_i} \cdot \mu_{N_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) \\
 \mu_{23} &= (\text{offset} + \mu_{HR} \cdot \mu_N) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SR_i} \cdot \mu_{N_i}) (1 - \mu_{SR_i} \cdot \mu_{V_i}) (1 - \mu_{HR_i} \cdot \mu_{F_i}) \\
 \mu_{24} &= (\text{offset} + \mu_{HR} \cdot \mu_F) \prod_{i=1}^{\#ofObstacle} (1 - \mu_{SR_i} \cdot \mu_{V_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i}) (1 - \mu_{HR_i} \cdot \mu_{F_i}) \\
 \mu_{25} &= (\text{offset} + \mu_{HR} \cdot \mu_{VF}) \prod_{i=1}^{\#ofObstacles} (1 - \mu_{SR_i} \cdot \mu_{F_i}) (1 - \mu_{HR_i} \cdot \mu_{V_i})
 \end{aligned} \tag{3}$$

**2.4 Defuzzification**

최종적으로 50-법칙 P/N 퍼지 제어기의 출력 값은 수식 (4)에 의해 결정된다.

$$\begin{cases}
 y = \frac{w_{HL} \cdot HL + w_L \cdot L + w_{SL} \cdot SL + w_S \cdot S}{w_{HL} + w_L + w_{SL} + w_S} & (w_{HL} + w_L + w_{SL} \geq w_{HR} + w_R + w_{SR}), \\
 y = \frac{w_{HR} \cdot HR + w_R \cdot R + w_{SR} \cdot SR + w_S \cdot S}{w_{HR} + w_R + w_{SR} + w_S} & (w_{HL} + w_L + w_{SL} < w_{HR} + w_R + w_{SR})
 \end{cases} \tag{4}$$

위 식의 각 출력 퍼지 집합에 대한 적합도는 다음의 수식 (5)에 의해 결정된다.

$$\begin{aligned}
 \mu_{HL} &= \mu_1 + \mu_2 + \mu_6, \\
 \mu_L &= \mu_3 + \mu_4 + \mu_7 + \mu_8, \\
 \mu_{SL} &= \mu_5 + \mu_9, \\
 \mu_S &= \mu_{10} + \mu_{11} + \mu_{12} + \mu_{13} + \mu_{14} + \mu_{15} + \mu_{20}, \\
 \mu_{SR} &= \mu_{19} + \mu_{25}, \\
 \mu_R &= \mu_{17} + \mu_{18} + \mu_{23} + \mu_{24}, \\
 \mu_{HR} &= \mu_{16} + \mu_{21} + \mu_{22}
 \end{aligned} \tag{5}$$

**3. Lookup-Table based design**

본 논문 2 장에서 퍼지 제어기에는 멤버함수로 가우시안 곡선을 가지는 함수를 사용한다. 하지만 이 함수 값은 임베디드 프로세서에서 처리하기 복잡한 실수의 곱셈이 포함되어 있다. 이러한 연산이 퍼지제어기의 연산속도를 떨어뜨려 실시간의 빠른 장애물 회피를 힘들게 하는 원인이 될 수도 있다. 따라서 본 장에서는 검색테이블을 사용하여 해당 실수연산의 결과를 얻는 방법에 대해 설명을 한다.

본 논문은 로봇의 장애물 회피를 위하여 복잡한 연산을 지양하고 가능한 단순한 구조의 제어 방법을 채택하였고, 임베디드 프로세서에도 적용 가능한 제어를 설계하는 것을 목적으로 하였다. 로봇의 장애물 회피를 위해서는 회피를 위한 정확한 판단도 중요하지만, 실시간 장애물회피를 가능하게 하기 위하여 가능한 빠른 장애물회피가 가능하도록 제어를 설계하고자 한다.

검색테이블 기반 퍼지제어기의 가장 기본적인 개념은 멤버함수가 정해져 있고 그 입력값의 범위가 정해져 있으므로 각 입력값에 대한 출력값 또한 미리 계산하여 정해 놓을 수 있으므로 그 값을 검색테이블에 저장해 놓고 사용하면 복잡한 연산을 피할 수 있다는 점이다. 본 논문에서 각도에 대한 입력값은 -90°에서 90°까지의 각도 범위에서 0.5°도 단위로 입력이 들어오므로 가능한 입력값은 총 361 개인데 각 방향에 관계된 멤버함수에 대해서 361 개의 모든 입력값이 필요한 것이 아니라 각 함수의 입력 값의 범위를 정하고 그 입력 범위에 해당하는 값만을 검색 테이블에 저장하면 된다. 퍼지 입력인 4 가지 종류 입력값에 대한 검색테이블의 예가 Fig. 5 에 나타나 있다. 각 검색테이블은 중심(c)이 0 인 함수들의 검색 테이블이다. 퍼지제어기의 모든 멤버함수들은 우함수이므로 중심을 기준으로 한쪽 방향의 값들만 저장하고 반대 방향의 값은 함수의 대칭성을 이용하여 얻으면 된다. 각도에 대한 멤버함수의 검색테이블에서 인덱스(index)가 매우커지면 출력값이 0 으로 수렴하므로, 입력이 204 를 초과할 경우에는 출력값을 0 으로 둔다. 따라서 검색테이블의 총 항목수는 205 개가 된다. 나머지 멤버 함수들의 검색 테이블도 비슷한 방식으로 정의할 수 있다. 이러한 방식은 연산을 별도로 하지 않고 검색테이블의 입력값에 대한 출력값이 함수의 입력값에 대한 출력값이 되게 하

Address	Data (4bytes)	Address	Data (4bytes)	Address	Data (4bytes)
0	1	0	1	0	1
1	0.999614271	1	0.999999777	1	0.999999777
⋮	⋮	⋮	⋮	⋮	⋮
102	0.018063013	4235	0.018581188	424	0.018407037
⋮	⋮	⋮	⋮	⋮	⋮
204	0.000000106	8469	0.000000119	847	0.000000119

Fig. 5 Lookup-table (LUT) buffers of full LUT based architecture

부분의 임베디드 프로세서는 한정된 메모리 자원을 가지고 있고, 이러한 메모리 자원을 프로그램 메모리 공간, 스택(stack) 공간, 인터럽트 루틴(interrupt routine) 등의 다양한 용도로 사용하게 된다. 따라서 많은 공간을 검색테이블이 사용하는 것은 시스템에 부담이 되며, 전체적인 시스템의 비용을 증가시킨다. 따라서 본 논문에서는 검색테이블 공유기법을 제안함으로써 이러한 문제점을 해결하고자 한다. 공유되는 공용의 검색테이블은 수식 (6)와 같은 기본 함수에 대한 값을 가진다.

$$f_{basis} = e^{-\frac{x_{trans}^2}{2\sigma_{BasisFunction}^2}} \quad (6)$$

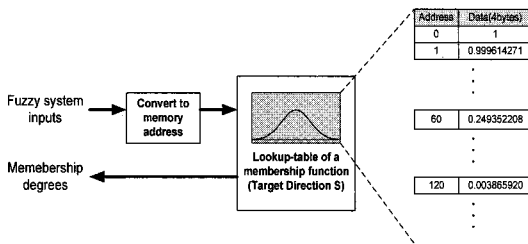


Fig. 6 Diagram of fuzzy membership degree acquisition on lookup-table based architecture

가우시안 곡선은 분산( $\sigma$ )과 중심( $c$ )으로 그 특징이 결정되며, 이 기본함수의 분산( $\sigma_{BasisFunction}$ )은 128 로 중심( $c_{BasisFunction}$ )은 0 으로 설정하였다. 그리고 퍼지제어기의 입력에 대한 각각의 멤버함수들의 값은 위의 기본함수의 입력 값을 아래의 수식 (7)을 이용하여 변경함으로써 얻을 수 있다.

므로 퍼지 제어기의 전체 연산 시간에서 멤버함수의 연산으로 인한 수행시간을 줄일 수 있다. Fig. 6 에 이러한 방식으로 멤버함수의 값을 얻는 예시가 있다. 여기서 입력 값은 검색 테이블이 존재하는 메모리상의 물리적인 주소 값으로 변환되고 그 주소의 메모리에서 읽어 들인 값이 멤버함수의 값이 된다.

$$x_{Converted} = \frac{\sigma_{BasisFunction}}{\sigma_{MembershipFunction}} (x_{input} - c_{MembershipFunction}) \quad (7)$$

이러한 방법을 사용하여 제어기의 연산시간을 단축 시키면 더 빠른 주기의 장애물회피를 위한 제어가 가능하다. 따라서 로봇의 이동 속도가 빠르거나 갑작스럽게 장애물이 발견 되었을 경우 좀 더 안정적으로 장애물을 회피할 수 있다.

하지만 수식 (7)에서 계산된  $x_{Converted}$  는 공용 검색테이블의 주소 값으로 사용할 수 있는 값이 아닐 수도 있다. 메모리상에 존재하는 검색 테이블의 주소는 정수로 표시가 된다. 만약  $x_{Converted}$  가 정수가 아니라 소수점 이하의 실수일 경우에는 이 입력 값을 위한 정확한 함수 결과 값이 현재 검색테이블에 존재하지 않는다. 이런 경우에는  $x_{Converted}$  에 대한 실제의 출력 값과 가장 근접한 값을 출력으로 가지는 주소로 변경하는 데 그 방법은 수식 (8)과 같다.

#### 4. Lookup-Table sharing method

$$Address_{LUT} = \left\lfloor \text{round} \left( \frac{\sigma_{BasisFunction}}{\sigma_{MembershipFunction}} (x_{input} - c_{MembershipFunction}) \right) \right\rfloor \quad (8)$$

검색테이블을 사용하여 퍼지제어기를 구현하는 것은 좀 더 빠른 제어를 가능하게 한다는 점에서 큰 장점을 가진다. 하지만 시스템이 각 멤버함수의 검색테이블을 위한 메모리 공간을 가지고 있어야 된다는 점에서 단점이 있다. 만약 퍼지 제어기가 더 복잡해지고 더 많은 퍼지법칙을 사용하게 된다면 그만큼 더 많은 메모리 공간이 필요하게 된다. 이러한 메모리의 사용은 임베디드 프로세서를 사용한 구현에서는 큰 단점이 될 수 있다. 대

물론 수식 (8)과 같은 방식으로 검색테이블의 주소를 구하면, 얻고자 하는 원래의 값과는 조금 다른 결과 값을 얻게 되지만, 퍼지제어기는 엄격한 수학적인 모델링으로 시스템의 출력을 얻는 것이 아니라 직관적이고 개념적인 방법으로 출력을

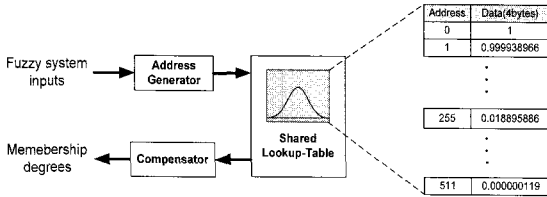


Fig. 7 Diagram of fuzzy membership degree acquisition on lookup-table sharing method

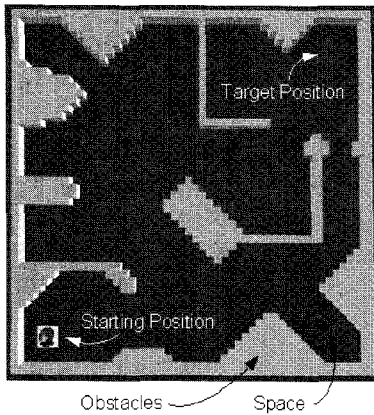


Fig. 8 Simulation map of MSRDS (Microsoft Robotics Developer Studio)

열는 시스템이므로 수식 (8)에서 발생하는 오차는 시스템 성능에 큰 영향을 미치지 않는다. 검색테이블로부터 값을 얻어낸 후 값을 Compensator 에서 보정해 줄 수 있지만, 이 또한 시스템 성능에 큰 영향을 주지 않고, 전체적인 연산만을 증가시키므로 제안한 제어기에는 사용을 하지 않았다.

Fig. 7 은 공유되는 공용의 검색 테이블을 사용하여 퍼지 멤버함수의 값을 구하는 과정을 나타낸다. 본 논문의 공용 검색테이블은 0 에서 511 까지의 인덱싱 주소(AddressLUT)를 가지고 있는데 각 메모리 주소에 32-bit 의 부동소수형(fixed-point) 데이터가 저장되어 있다. Address Generator 가 공용 검색테이블의 인덱싱 주소(AddressLUT)를 계산하는데, Table 1 에 나타난 각 멤버함수의 특성에 맞게 계산이 된다. 만약 계산된 주소가 511 을 초과하게 되면, 메모리의 검색 없이 0 이 멤버함수의 값이 된다. 가우시안 곡선 형태의 함수는 입력 값이 중심에서 멀어질수록 출력 값이 0 에 가까워지는데, 그 값이 매우 작은 출력 값은 퍼지 제어기에 거의 영향을 주지 않으므로 0 으로 설정함으로써 메모리 사용량을 줄일 수 있다. 본 논문의 공용

Table 5 Buffer size and processing cycles of each fuzzy controller

	calculating w/o LUT	using full LUT	sharing a LUT
Buffer size of Lookup-table(bytes)	0	38,080	2,048
Clock cycles per 1 avoidance (cycles)	427,390	84,797	127,494

검색테이블에서는 0.000000119 미만의 값은 0 으로 간주한다.

### 5. Results of simulation and experiment

본 논문에 제안된 50-Rule P/N 퍼지 제어기는 로봇 통합 개발 도구 중 하나인 MSRDS 에서 시뮬레이션이 수행되었다. 시뮬레이션이 수행된 컴퓨터 시스템은 Intel® Core™ 2 Quad 2.4GHz CPU 에 3.24GB RAM 의 사양을 가지고 있다. Fig. 8 은 MSRDS 시뮬레이션 환경에서 장애물회피 로봇의 공간적인 지도이다. 이 지도에서 로봇은 출발점 (Starting Position)에서 목표점(Target Position)까지 주행을 하는데, 만약 두 점 사이에 아무런 장애물이 없다면 로봇은 그대로 직진하여 목표점에 도달하게 된다. 그림의 시뮬레이션 환경에서는 여러 미로와 같은 장애물들이 존재하는데, 이러한 정보는 로봇이 사전에 알고 있지 않고, 주행 중에 센서를 통해 장애물 정보를 입력 받은 것으로 설정하였다. 즉 로봇은 주행 중 입력 받은 장애물 정보를 가지고 장애물을 회피하며 목표점으로 주행하게 된다. 제안된 방법을 시뮬레이션 하기 위하여 C# 프로그래밍과 Microsoft Visual Studio 컴파일러(compiler)를 사용하여 MSRDS 서비스 모듈을 생성하였다. 또한 다른 구조와의 비교를 위하여 검색테이블을 사용하지 않은 방법, 모든 멤버함수에 대한 전역 검색테이블을 사용하는 방법에 대해서도 서비스 모듈을 생성하였다. 그리고 VPL(Visual Programming Language) 도구를 이용하여 생성한 각 서비스 모듈을 적용하고 시뮬레이션을 수행하였다. 시뮬레이터에서 로봇의 센서 및 구동과 관련된 서비스 모듈은 MSRDS 에서 라이브러리(library)로 제공되는 것을 사용하였다. 수행된 시뮬레이션에는 그래픽 기반 환경이 적용되므로 고성능 컴퓨터 시스템을 사용하였다. 그래픽기반 시뮬레이터에서는 컴



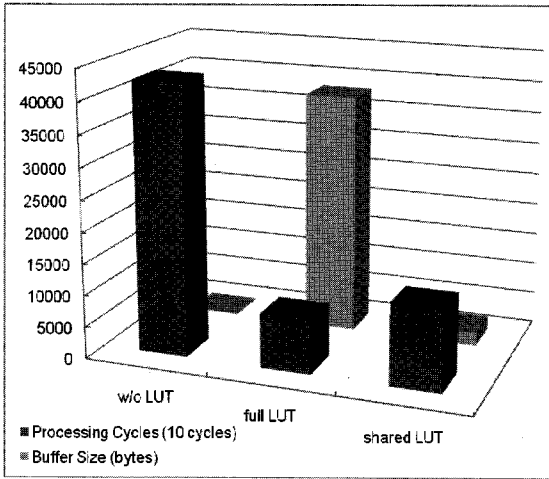


Fig. 9 Comparison of buffer size and average processing cycles

퓨팅 시간의 대부분을 그래픽작업에 소요하기 때문에 퍼지 제어기의 성능을 객관적으로 비교하기 위해서는 시뮬레이션 과정에서 퍼지 제어기 서비스 모듈에서 사용한 시간만을 측정하여 비교하는 것이 바람직하다.

Table 5 와 Fig. 9 는 수행한 3 가지 방법에 대한 시뮬레이션 결과를 나타낸다. 전역 검색테이블 기반 구조에서의 버퍼 크기는  $(204+8469+847) \times 4$  bytes 이며 Fig. 7 과 같이 검색테이블 공유 기법을 사용한 구조의 버퍼 크기는  $512 \times 4$  bytes 이다. 그림에서의 연산 주기(processing cycles)는 목표점까지의 주행 시뮬레이션에서 한번의 회피 동작에 퍼지 제어기가 평균적으로 사용하는 연산 수행클럭수(processing clock cycles)를 의미한다. 검색테이블을 사용하지 않는 구조는 멤버함수를 위한 버퍼 메모리가 필요하지 않지만 전역 검색테이블 기반 구조보다 8 배 가량 평균 연산주기가 긴 것으로 나타났다. 하지만 전역 검색테이블 기반 구조는 38 Kbytes 의 메모리가 필요한데 이는 소형 임베디드 프로세서에서는 부담이 될 수 있다. 이를 위해 공용의 검색테이블 공유기법을 사용하면 2 Kbytes 의 메모리만으로 검색테이블 기반 퍼지제어기를 구현할 수 있다. 이 방법을 사용하면 전역 검색테이블 기반 제어기보다 1.5 배 정도 연산 주기가 길어지지만 버퍼 메모리 크기를 94%이상 줄일 수 있으므로 유용하게 사용될 수 있다. 만약 시스템이 충분한 용량의 메모리 가지고 있으면 전역 검색테이블 기반 구조가 더 유용하며, 검색테이블 공유기

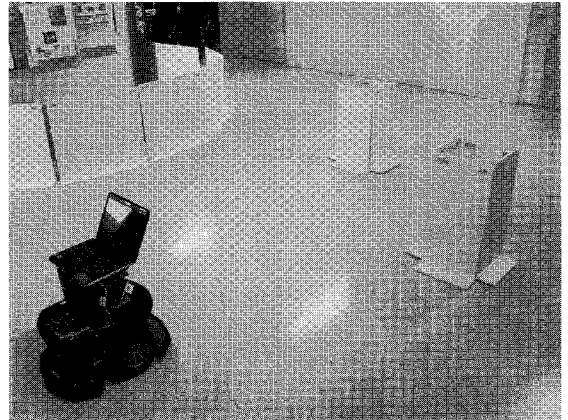


Fig. 10 The experimental environment: LabVIEW applications on the mobile robot pioneer 3-DX

법은 메모리 사용이 제한적인 시스템에서 유용하게 사용될 수 있다.

시뮬레이션 결과를 요약하자면 다음과 같다. 제안된 퍼지제어기에 검색테이블 기반 구조를 사용하면 상당한 연산 수행 주기를 줄일 수 있으며, 검색테이블 공유기법을 사용했을 때에는 메모리 사용량을 크게 줄일 수 있어 시스템 메모리의 큰 부담 없이 검색테이블을 사용할 수 있다는 것을 알 수 있다. 최종적으로 제안한 구조의 퍼지제어기를 통합 개발도구인 NI사의 LabVIEW 를 사용하여 구현하였고 이를 실제의 실험으로 동작을 검증하였다. Fig. 10 에 실제 모바일 로봇을 이용한 장애물회피 실험환경이 나타나 있는데 우측상단에 노란색 사각형이 있는 지점이 주행의 목표점이 된다. Fig. 11 은 실험환경에서 해당하는 장애물을 회피하여 주행한 결과로써의 경로를 나타낸다.

## 6. Conclusions

본 논문은 모바일 로봇의 실시간 장애물회피에 대해 다룬다. 안정적인 장애물회피를 위하여 독립적인 프로세서에서의 제어기 구현을 목표로 하고 있으며, 반복적이고 빠른 연산이 가능한 제어기 구조를 제안한다. 모바일 로봇 시스템에서 장애물 회피를 위한 전용 프로세서를 운용하기 위해서는 시스템 비용과 전력소모를 고려했을 때 임베디드 프로세서를 이용하는 것이 적절하다. 이를 위하여 본 논문에서는 비교적 적은 시스템 자원을 활용하면서도 빠른 연산을 가능하게 하는 검색테이블 공

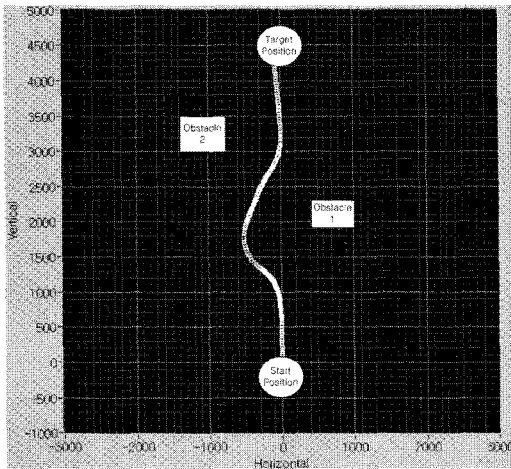


Fig. 11 The experimental result of obstacle avoidance using proposed fuzzy controller

유기법을 제안하고 시뮬레이션 및 실험을 수행하였다. 검색테이블 공유기법은 반복적이고 복잡한 멤버함수 연산을 검색테이블 기반으로 구현하는데, 메모리의 사용량을 크게 줄여 저비용의 임베디드 프로세서에서도 실시간 장애물회피 제어를 적용할 수 있도록 하는 데 의미가 있다.

### 후 기

본 연구는 교육과학기술부 일반사업 연구비 지원에 의해 수행되었습니다.

### 참고문헌

1. Kim, J., Kim, Y., Kang, W. and An, J., "Design of a Lookup-table Based Positive/Negative Fuzzy Controller for Obstacle Avoidance of Autonomous Vehicle," 3<sup>rd</sup> ICUT, pp. 648-652, 2008.
2. Koren, Y. and Borenstein, J., "Potential field methods and their inherent limitations for mobile robot navigation," IEEE Int. Conf. Robotics and Automation, pp. 1398-1404, 1991.
3. Borenstein, J. and Koren, Y., "The vector field histogram-fast obstacle avoidance for mobile robots," IEEE Int. Conf. Robotics and Automation, pp. 278-288, 1991.
4. Thongchai, S. and Kawamura, K., "Application of fuzzy control to a sonarbased obstacle avoidance mobile robot," IEEE Int. Conf. Control Applications,

pp. 425-430, 2000.

5. Wang, L.-X., "Adaptive Fuzzy Systems and Control: Design and Stability Analysis," Pientice-Hall, 1994.
6. Palm, R., Driankov, D. and Hellendoorn, H., "Model Based Fuzzy Control: Fuzzy Gain Schedulers and Sliding Mode Fuzzy Controllers," Springer-Verlag 1997.
7. Passino, K. M. and Yurkovich, S., "Fuzzy Control," Addison Wesley Longman, 1998.
8. Aguirre, E. and Gonzalez, A., "Fuzzy behaviors for mobile robot navigation," International Journal of Approximate Reasoning, Vol. 25, No. 3, pp. 255-289, 2000.
9. Mbede, J. B., Ma, S., Zhang, L., Toure, Y. and Graefe, V., "Robust neuro-fuzzy navigation of mobile manipulator among dynamic obstacles," IEEE Int. Conf. on Robotics and Automation, Vol. 5, pp. 5051-5057, 2004.
10. Lilly, J. H., "Evolution of a Negative-Rule Fuzzy Obstacle Avoidance Controller for an Autonomous Vehicle," IEEE Trans. on Fuzzy Systems, Vol. 15, No. 4, pp. 718-728, 2007.
11. Kiendl, H., "Design of advanced fuzzy systems," Proc. TOOLMET'99 Symp.-Tool Environ. Develop. Meth. Intell. Syst., 1999.
12. Branson, J. S., "A Trainable fuzzy system incorporating dynamically variable asymmetric spreads and negative rule defuzzification," Ph. D. Dissertation, Dept. of Electrical and Computer Engineering, Univ. of Louisville, 2000.
13. <http://msdn.microsoft.com/en-us/robotics>