

# 지능로봇: 지능 에이전트를 기초로 한 접근방법

## Intelligent Robot Design: Intelligent Agent Based Approach

강진식  
Jin-Shig Kang

제주대학교 공과대학 통신컴퓨터 공학부  
Dept. Telecommunication and Computer Eng. Cheju Nat'l Univ.

### 요 약

본 논문에서는 로봇을 하나의 에이전트로 보고 로봇에 필요한 기능-환경인지, 지능, 행동-등을 부-에이전트로 하는 로봇 구조를 제안 하였다. 각각의 부 에이전트들은 로봇 기능의 기초단위를 구성하는 마이크로 에이전트로 구성된다. 로봇의 제어 구조는 행위기반 반사행동 제어 형태와 행동 선택 에이전트로 구성되며, 행동 선택 에이전트에서의 행동 선택은 행동 우선순위, 수행성능, 강화학습에 의한 학습기능을 부가하였다. 제시된 로봇 구조는 다중 부-에이전트 구조로 각각의 기능에 대하여 지능을 부여하기 쉬우며 다중 로봇 제어를 위한 새로운 접근 방법이다. 제시된 로봇을 장애물을 회피와 chaotic한 탐색을 목표로 하여 모의실험을 수행하였으며 8bit 마이크로 콘트롤러를 이용하여 제작 실험하였다.

### Abstract

In this paper, a robot is considered as an agent, a structure of robot is presented which consisted by multi-subagents and they have diverse capacity such as perception, intelligence, action etc., required for robot. Also, subagents are consisted by micro-agent( $\mu$ agent) charged for elementary action required. The structure of robot control have two sub-agents, the one is behavior based reactive controller and action selection sub agent, and action selection sub-agent select a action based on the high label action and high performance, and which have a learning mechanism based on the reinforcement learning. For presented robot structure, it is easy to give intelligence to each element of action and a new approach of multi robot control. Presented robot is simulated for two goals: chaotic exploration and obstacle avoidance, and fabricated by using 8bit microcontroller, and experimented.

**Key words** : 지능 에이전트, 지능 로봇, 반사행동 제어기, 강화학습, 부-에이전트

### 1. 서 론

최근 지능형 로봇에 대한 관심이 높아지고 있으며 이에 대한 연구가 활발히 진행되고 있다. 로봇 제어에 관한 연구는 제어 이론에 기초한 모델기반 제어로 과거의 로봇-암 제어에서 주로 사용되던 방법과 지능형 제어로서 반사행동 제어인 행위기반 제어, 모터 스키마 제어 등의 방법이 연구되어 왔으며 모델기반 제어와 행위기반 제어를 조합한 하이브리드 제어에 대한 연구도 있다. 행위기반 반사행동 제어는 R. A. Brooks[1]의 subsumption 알고리즘이 그 시초로 최근 많은 연구가 이 분야에 집중되어지고 있다[1][2][3]. 행위기반 제어구조는 고전적인 수직적 제어구조가 아니라 수평적 제어구조로 목표들이 충돌 이루며 병렬 처리에서와 같은 방법으로 행위를 위한 연산을 수행한다. 이 결과는 subsumption 알고리즘에 의하여 선택 된다. 이러한 수평적 제어구조는

응답시간이 짧으며 제어 알고리즘의 구현이 매우 쉬운 장점을 갖는다. 그러나 목표와 이동로봇의 경로 계획이 어려우며 목표들간의 상호 작용에 의한 진동현상(e.g., limit cycle)이 일어날 수 있다. 이러한 문제를 해결하기 위한 방법으로 포테셜 필드 방법[4-9]과 모터 스키마를 이용한 계획법[10-11]이 연구되었다. 또 다른 방법으로는 학습에 의한 제어 방법[12-13]과 하이브리드 제어 방법[14] 등이 있다.

본 논문에서는 로봇을 하나의 에이전트로 보고 로봇에 필요한 기능인 환경인지, 지능, 행동등을 부-에이전트로 하는 로봇 구조를 제시 한다. 각각의 부 에이전트들은 로봇 기능의 기초단위를 구성하는 마이크로 에이전트로 구성되며 로봇의 제어 구조는 행위기반 반사행동 제어 형태와 행동 선택 에이전트로 구성된다. 행동 선택 에이전트에서의 행동 선택은 행동 우선순위, 수행성능, 강화학습에 의한 학습기능을 부가한다. 제시된 로봇 구조는 다중 부-에이전트 구조로 각각의 기능에 대하여 지능을 부여하기 쉬우며 다중 로봇 제어를 위한 새로운 접근 방법으로 장애물을 회피와 chaotic한 탐색을 목표로 하여 모의실험을 수행한다.

접수일자 : 2004년 6월 15일  
완료일자 : 2004년 6월 24일

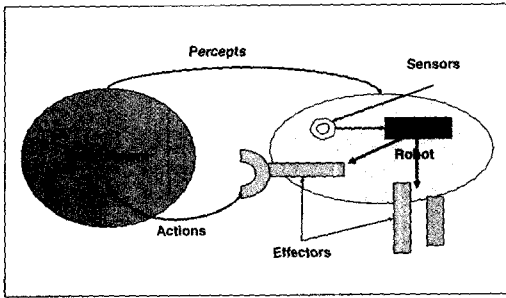


그림 1. 에이전트의 기본구조  
Fig. 1. Basic structure of agent

## II. 지능형 에이전트

### 2.1. 지능형 에이전트

에이전트는 다른 사람의 권리와 권한을 행하는 사람, 또는 위임 받은 사람이라는 사전적 의미를 지니고 있지만 최근 공학에서는 주변환경을 인지하고 그 환경에 따르는 행동을 결정하며 그 행동을 수행하는 물리장치 혹은 소프트웨어로 취급되고 있다. 그림 1은 에이전트에 대한 기본 구조이다. 에이전트는 기본적으로 그 에이전트가 행동할 수 있는 공간인 환경과 환경을 인지(perception)할 수 있는 센서와 행동을 결정하기 위한 지능, 그리고 행동을 표현할 수 있는 작동기 등으로 구성된다. 환경은 이미 알려져 있으며 불확정성(uncertainty)이 없는 구조화(structured)된 환경, 알려져 있지 않으며 불확정성을 포함하는 비 구조화(un-structured)된 환경, 그리고 부분적으로 알려져 있으며 불확정성을 부분적으로 포함하는 부분 구조화(partially structured)된 환경으로 나눌 수 있다. 대부분의 로봇이 어떤 목표를 수행하는 환경은 부분적으로 구조화 되거나 구조화 되지 않은 환경이 대부분이며 이에 따라 자율적으로 목표를 정하고 그 목표를 오퍼레이터의 도움이 없이도 수행할 수 있어야 한다. 따라서 로봇(에이전트)은 지능을 가지고 있어야 하며 환경에 적응하기 위한 학습 능력을 갖추고 있어야 한다. 지능형 에이전트가 가져야 하는 특징들은 다음과 같다.

- (1) 자율성(Autonomy):
- (2) 반사행동(Reactivity):
- (3) Pro-activeness:
- (4) 사회성(Social Ability):
- (5) 합리성(Rationality):
- (6) 다양성(Versatility):
- (7) 적응성(Adaptability):
- (8) 이동성(Mobility):

### 2.2. R. Brooks의 Subsumption 구조: 행위기반 로봇

과거 로봇을 제어하기 위한 제어 구조로는 그림 2으로 설명될 수 있는 계층제어구조를 주로 사용하여 왔다. 이 제어구조는 로봇의 행동 선택이 수직적이며 순서적이어서 로봇의 지식을 표현하거나 다양한 형태의 목표를

설정하기가 어렵다. 로봇이 다양한 기능을 하기 위해서는 복잡한 알고리즘을 구현하여야 하며 이러한 경우는 응답속도가 아주 느려지게 된다. 이와는 반대로 그림3의 subsumption 제어구조는 수직적으로 배열된 여러 목표가 층을 이루고 있으며 각 층은 독립적으로 로봇의 행동을 계산한다. 목표는 그 중요도에 따라 각 층에 할당되며 중요도가 낮으면 아래층에 높으면 상위 층에 배열한다. 그리고 각 층에서 계산된 로봇의 행동은 subsumption 알고리즘에 의하여 선택 된다. 즉, 상위 층은 하위 층의 행동을 포함하며 상위 층에 의하여 계산된 행동은 아래층에 의하여 계산되는 모든 행동을 포함한다.

R. Brooks의 subsumption 제어 구조는 행위기반 (behavior based) 로봇 제어의 토대가 되었으며 최근 많은 연구가 이루어 지고 있다.

### 2.3. 문제서술

최근 행위기반 로봇 제어문제, 계층적 로봇 제어문제, 그리고 로봇에 지능을 부여하고 학습을 통하여 지능을 축적하는 문제에 대한 많은 연수가 수행되어 왔지만 제어가 쉬운 로봇 설계에 대한 연구는 등한시 되어 왔다. 따라서 구조가 단순하면서도 지능을 가지고 학습할 수 있는 로봇이 요구되고 있다.

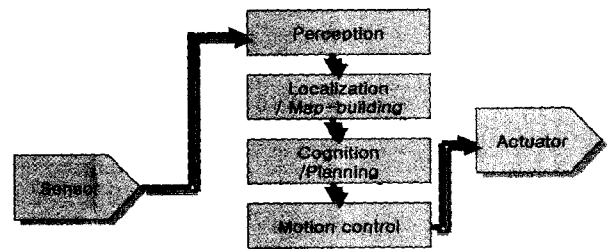


그림 2. 계층적 제어구조  
Fig. 2. The hierarchical control structure

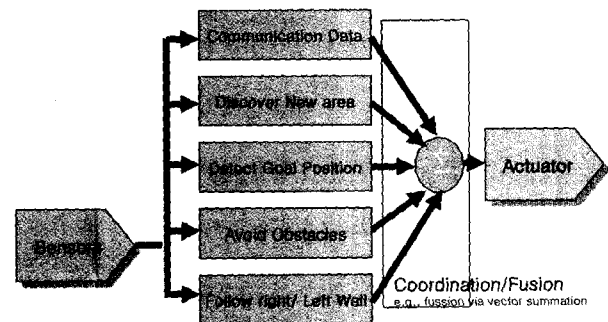


그림 3. R. Brooks의 subsumption 제어구조  
Fig. 3. R. Brooks's subsumption control structure

본 논문에서는 구조가 간단하며 제어가 쉬운 지능형 로봇을 제시한다. 제시된 로봇의 각각의 기능을 향상 시키기 위하여 로봇을 하나의 에이전트로 보고, 이 에이전트는 다양한 부-에이전트와 마이크로 에이전트로 구성한다. 각각의 부-에이전트와 마이크로 에이전트를

정의하며 에이전트 그룹, 매크로 에이전트 등을 정의한다. 제시된 구조로 로봇을 설계하고 이 로봇을 이용하여 장애물 회피와 탐색을 목표로 한 모의실험을 수행 하며 실험을 통하여 장애물을 회피할 수 있음을 보인다.

### III. 에이전트기반 지능로봇

이 장에서는 지능형 로봇의 설계 방법에 대하여 서술한다. 지능형 로봇을 설계하는 과정으로 반사행동 제어 형태를 고려하여야 하지만 이 제어 방식은 학습기능이 뒤떨어진다. 이를 극복하기 위하여 에이전트의 다양한 형태를 정의하며 각 에이전트들의 기능을 서술한다.

#### 3.1. 에이전트 정의

에이전트는 지각기능, 학습과 행동기능, 환경 등의 요소로 가진 물리 장치, 혹은 소프트웨어이다. 그러나 로봇처럼 다양한 행동이 요구되며, 다양한 센서들로 구성된다면 단순한 형태의 에이전트에 대한 정의만으로 학습을 포함하는 로봇의 모든 기능을 설명하기에 부족하다. 또한 최근의 로봇 연구는 다중-로봇을 제어하기 위한 알고리즘에 집중되어 있으며 이를 다중에이전트로 정의하여 제어하려는 시도들이 있었다. 본 논문에서 새롭게 정의되는 에이전트들은 다음과 같다.

- **부-에이전트(sub-Agent):** 에이전트의 요소-지각, 학습, 행동들 중에 하나 또는 하나 이상의 요소로 구성되는 에이전트.
- **마이크로 에이전트(Micro Agent):** 독립된 하나의 센서, 하나의 모터와 같이 에이전트를 구성하는 최 하위 인자.
- **매크로 에이전트(Macro Agent):** 하나 이상의 에이전트로 구성되며 같은 감독에이전트(supervisor agent)의 통제를 받는다.
- **에이전트 그룹(Agent Group):** 하나 이상의 에이전트로 구성되며 동일한 전술을 수행하지만 서로 다른 감독 에이전트의 통제를 받을 수 있다.

#### 3.2. Robot Structure

그림 4는 본 논문에서 제시하는 로봇구조 이다. 로봇은 센서 부-에이전트, 행동계층, 행동선택 부-에이전트, 행동 부-에이전트, 목표계획 부 에이전트, 통신 부-에이전트로 구성되며 각각의 부-에이전트 들은 하위의 마이크로 에이전트 들로 구성된다.

##### (1) 센서 부-에이전트

센서 부-에이전트는 마이크로 에이전트인 sensor 0, ..., sensor k와 각 목표(goal)에 필요한 센서신호를 보내기 위한 선택부분으로 구성되며 각 마이크로 에이전트는 센서 신호로부터 환경을 인식하기 위한 학습 알고리즘으로 구성된다.

##### (2) 목표계층

목표는 층을 이루고 있으며 각 층간의 행동 연산은 서로 독립적이다. (Goal i) 목표 계층은 센서 신호와 목표계획 부-에이전트로부터 변경된 목표를 입력으로 받으며 독립적으로 로봇의 행동을 결정하여 출력한다. 이때 센서 신호는 센서 부-에이전트로부터 (Goal i) 목표 계층에 필요한 신호만을 받는다. 각각의 목표 계층들은 중요도에 따라 상위 계층과 하위계층으로 구성되며 중요도 평가 지수인 수행성능 값을 출력할 수 있다.

##### (3) 행동선택 부-에이전트

행동선택 부-에이전트는 각 목표계층의 행동들과 목표계획 부-에이전트로부터 성능평가지표를 입력으로 받으며 각 목표계층의 출력들과 성능 값, 그리고 성능 평가 지표를 이용하여 행동을 선택한다. 선택된 행동은 행동 부-에이전트로 전달되어 행동 하도록 한다.

##### (4) 행동 부 에이전트

행동 부 에이전트는 액추에이터를 구동하기 위한 마이크로 에이전트들과 상과 벌을 결정하기 위한 마이크로 에이전트로 구성되며 이 부-에이전트는 행동의 적합성을 판단하여 상(reward)과 벌(penalty)의 값을 부여하고 그 값을 행동 선택 부-에이전트로 전달한다. 이 방법은 강화 학습(reinforcement learning)으로 동적으로 변화하는 환경에 대하여 비교적 강인한 것으로 알려졌다.

##### (5) 목표계획 부-에이전트

목표계획 부-에이전트는 감독-에이전트와 에이전트 간의 통신을 담당하는 통신 부 에이전트로부터 명령을 입력 받으며 행동 부-에이전트로부터 행동과 행동에 대한 평가를 입력으로 받아 새로운 명령을 계획하고 수행성능 평가지수와 새로운 목표를 출력 한다.

##### (6) 통신 부 에이전트

통신 부 에이전트는 에이전트간의 통신과 에이전트와 감독 에이전트간의 통신을 담당한다.

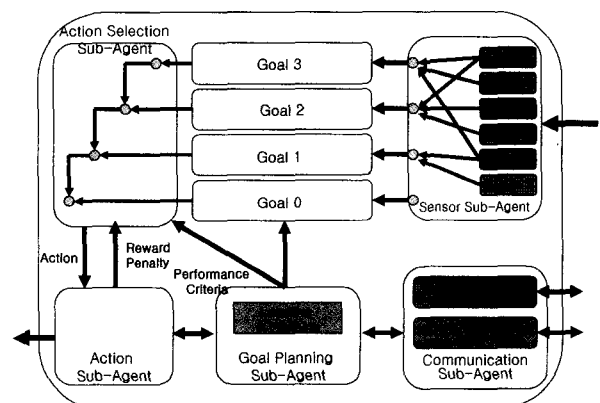


그림 4. 로봇의 구조  
Fig. 4. Robot structure

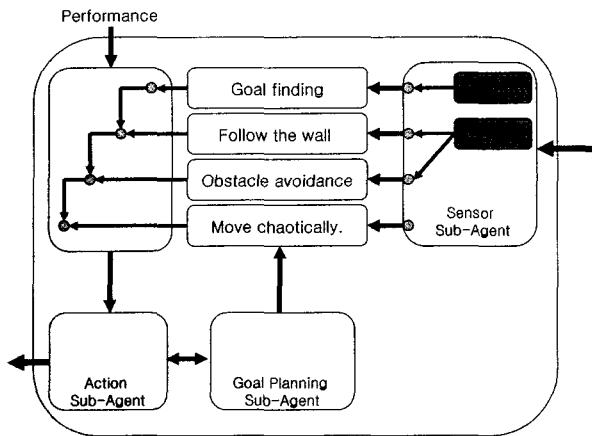


그림 5. 로봇 제어구조  
Fig. 5. Robot control structure

3.3. 로봇설계

본 논문에서는 통신 부-에이전트를 제외하며 다음의 5가지 행동 목표를 갖도록 하였다.

- Goal 0: Chaos하게 이동하며 목표를 탐색한다.
- Goal 1: 장애물이나 벽을 만나면 위치 좌표를 기록한다.
- Goal 2: 벽을 만나면 벽을 따라 목표를 탐색한다.
- Goal 3: 장애물을 만나면 피한다.
- Goal 4: 목표를 만나면 정지한다.

제어기 구조는 그림 5와 같다.

(1) 센서 부-에이전트

위의 5가지 행동목표를 성취하기 위해서 센서로는 장애물을 인식하기 위한 센서, 목표점을 확인할 수 있는 센서, 그리고 주행경로를 확인하기 위한 주행기록 센서들이 필요하다. 장애물을 검색하기 위한 센서는 초음파 센서와 적외선 센서가 고려될 수 있지만 모의실험에서는 측정범위가 15cm, 측정 각도가 45°인 초음파 센서를 가정하였다. (초음파 센서의 감지거리는 15cm보다 크지만 로봇의 행동에 영향을 미치는 감지거리는 15cm로 설정하였다.) 이때 초음파의 이동거리는 30cm 미만으로 시간은 8.823ms이다. 장애물을 회피하기 위한 초음파 센서는 그림 6과 같이 8방향으로 설정하였다. 이 센서들은 로봇의 각도와 위치가 지정된 범위 내에 들어오면 1이고 그렇지 않은 경우에는 0이 된다. 목표를 탐색하기 위하여 4개의 적외선 센서를 사용하며 로봇이 목표점 위에 위치하면 4개의 센서가 1이 되도록 설정하였다. 위치좌표를 기록하기 위하여 사용되는 센서는 잡음과 오차에 의한 영향이 작아야 하며 측정된 오차를 보정할 수 있어야 한다. 오차의 보정은 센서에 벽이 감지될 경우 벽의 좌표 값을 이용하여 보정할 수 있다. 센서 부 에이전트의 유사코드는 다음과 같다.

```
Function: Sensor-Sub-Agent(sensor data)
Input: sensor data
State:
Function: Side Sensor-μ -Agent(sensor data)
```

```
Function: Bottom Sensor-μ -Agent(sensor data)
Function: Odometer Sensor-μ -Agent(sensor data)
Output: sensor data
```

```
Function: Side Sensor-μ -Agent(sensor data)
Input: sensor data
State: i=1:8
S(i)=1←T<8.823ms
S(i)=0←otherwise
Output: sensor data S
```

```
Function: Bottom Sensor-μ -Agent(sensor data)
Input: sensor data
State: i=1:4
Sb(i)=1←T<8.823ms
Sb(i)=0←otherwise
Output: sensor data Sb
```

```
Function: Odometer Sensor-μ -Agent(sensor data)
Input: sensor data
State: i++
SR++, SL++
i=iMax
SR←0, SL←0
Output: sensor data SR, SL
```

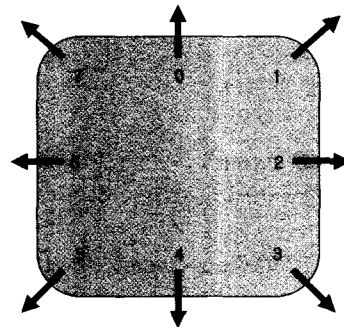


그림 6. 센서 방향  
Fig.6. Direction of Sensor

(2) 목표계층

각각의 행동은 센서 값에 따라 우측바퀴의 속도와 좌측 바퀴의 속도가 바로 출력될 수 있도록 하여야 한다. 제시된 로봇은 5가지의 행동으로 구성되며 각 행동은 우선순위에 따라 상위계층의 행동과 하위계층의 행동으로 구분된다.

목표탐색은 최 상위 계층이며 목표가 탐색되면 정지하여야 한다. 각 시간에서의 목표 탐색 결과는 0(성공) 또는 1(실패)을 출력하며 이 값은 하위 행동에서 결정되는 속도에 곱한다.

```
Function: Goal 4 (sensor data, Vr, Vl)
Input: sensor data
State: Goal←True
Vr←0, Vl←0
Goal←False
Vr, Vl
Output: velocity Vr, Vl
```

장애물 회피는 포텐셜 필드 방법(potential field method: PFM)과 벡터필드 방법(vector field method: VFM) 등이 주로 연구되어 왔지만 국소진동과 국소 최소(local minima)에 빠질 염려가 있다. 따라서 본 논문에서는 변형된 벡터필드 방법과 학습에 의하여 보정하는 새로운 방법을 제시한다.

장애물 회피 행동은 센서 신호를 이용하여 결정될 수 있다. 임의의 센서 신호가 1이면 로봇은 그 센서가 지정하는 반대 방향으로 움직이도록 하여야 하며 최종 행동은 각 행동의 합으로 표현될 수 있다. 예를 들어 그림 7과 같은 막다른 위치에 로봇이 위치하게 되면 4를 제외한 모든 센서는 1이 되며 이때 로봇이 진행 하여야 하는 방향은 4방향이다. 즉 모든 반사 힘의 벡터 합은 4가 된다. 이는 VFM과 유사한 방법으로 국소 최소에 빠질 수 있으며 좁은 통로를 통과하지 못한다. 이러한 문제를 해결하기 위하여 본 논문에서는 가중치가 부과된 새로운 장애물 회피 방법을 제시한다. 전진속도를 포함하는 로봇에 가해지는 힘의 총합은 다음과 같다.

$$F = F_f + F_0 + F_1 + F_2 + F_3 + F_4 + F_5 + F_6 + F_7$$

식 (1)에서 센서가 감지된 힘은 값을 갖지만 센서가 감지되지 않은 힘은 0이다. 식 (1)과 같이 합으로 구해진 힘은 로봇의 운동 속도로 변환되어야 하며 표 1에 변환을 위한 표를 보였다.

표 1. 운동속도  
Table 1. Velocity

F	V <sub>r</sub>	V <sub>l</sub>	F	V <sub>r</sub>	V <sub>l</sub>
F <sub>0</sub>	-V+δ <sub>0</sub>	-V+δ <sub>0</sub>	F <sub>4</sub>	V+δ <sub>4</sub>	V-δ <sub>4</sub>
F <sub>1</sub>	V+δ <sub>1</sub>	V-δ <sub>1</sub>	F <sub>5</sub>	V+δ <sub>5</sub>	V-δ <sub>5</sub>
F <sub>2</sub>	V+δ <sub>2</sub>	V-δ <sub>2</sub>	F <sub>6</sub>	V+δ <sub>6</sub>	V-δ <sub>6</sub>
F <sub>3</sub>	V+δ <sub>3</sub>	V-δ <sub>3</sub>	F <sub>7</sub>	V+δ <sub>7</sub>	V-δ <sub>7</sub>

표 1에서의 δ<sub>i</sub>는 센서의 각도에 의하여 구하여 진다. 식(1)의 전체 힘F를 표 1에 의하여 변환하면 로봇의 속도가 기준속도보다 커지기 때문에 정규화가 필요하다. 정규화를 위하여 벡터W를

$$W = [w_f, w_0, w_1, w_2, w_3, w_4, w_5, w_6, w_7]^T \quad (2)$$

$$\sum_i w_i = 1$$

라 하면 정규화된 힘 F<sub>n</sub>은

$$F_n = w_f F_f + w_0 F_0 + w_1 F_1 + w_2 F_2 + w_3 F_3 + w_4 F_4 + w_5 F_5 + w_6 F_6 + w_7 F_7$$

$$= V + w_0 \delta_0 + w_1 \delta_1 + w_2 \delta_2 + w_3 \delta_3 + w_4 F_4 + w_5 \delta_5 + w_6 \delta_6 + w_7 \delta_7 \quad (3)$$

와 같다. 이때 w<sub>i</sub>는 i번째 센서 신호가 1이면 값을 갖고 값을 갖지 않으면 0이다. 식 (3)을 로봇 좌표에 대하여 각 성분 별로 표시하면 다음과 같다.

$$\begin{bmatrix} {}^R F_x \\ {}^R F_y \end{bmatrix} = \begin{bmatrix} V w_f \cos \theta \\ V w_f \sin \theta \end{bmatrix} + \begin{bmatrix} V \sum_i w_i \cos \theta_i \\ V \sum_i w_i \sin \theta_i \end{bmatrix} \quad (4)$$

식 (4)에서 θ<sub>i</sub>는 센서 각도로서 [180°, -135°, -90°, -45°, 0°, 45°, 90°, 135°]이며 θ = 0° 이다. 따라서 로봇이 진행하여야 하는 방향 θ<sub>d</sub>는

$$\theta_d = \tan^{-1} \left( \frac{{}^R F_y}{{}^R F_x} \right)$$

$$= \tan^{-1} \left( \frac{\sum_i w_i \sin \theta_i}{w_f + \sum_i w_i \cos \theta_i} \right) \quad (5)$$

로 구하여 진다. 가중치를 계산하기 위하여 식 (5)를 변형하면

$$w_f \tan \theta_d - \sum_i w_i (\sin \theta_i - \tan \theta_d \cos \theta_i) = 0 \quad (6)$$

인 관계를 알 수 있다. 식 (6)은 벡터 내적으로 다음과 같이 쓸 수 있다.

$$0 = W^T S = [w_f, w_0, \dots, w_7] \begin{bmatrix} \tan \theta_d \\ \tan \theta_d \cos \theta_0 - \sin \theta_0 \\ \tan \theta_d \cos \theta_1 - \sin \theta_1 \\ \vdots \\ \tan \theta_d \cos \theta_7 - \sin \theta_7 \end{bmatrix} \quad (7)$$

식 (7)에서 w<sub>f</sub>와 w<sub>0</sub>는 서로 반대방향의 운동을 지시하며, 0번 센서가 감지되면 w<sub>f</sub> = 0, w<sub>0</sub> ≠ 0 이며 감지되지 않으면 w<sub>f</sub> ≠ 0, w<sub>0</sub> = 0 이다. i번째 센서만 감지되면 식 (7)은

$$0 = W^T S = [w_f, 0, \dots, w_i, \dots, 0] \begin{bmatrix} \tan \theta_d \\ 0 \\ \tan \theta_d \cos \theta_i - \sin \theta_i \\ \vdots \\ 0 \end{bmatrix} \quad (8)$$

이 되며 이때 가중치 w<sub>f</sub>, w<sub>i</sub>는 w<sub>f</sub> = 1로 두고 식 (6)을 이용하여 w<sub>i</sub>를 구하며 w<sub>f</sub> + w<sub>i</sub> = 1이 되도록 정규화 함으로서 계산할 수 있다. 동시에 두 개 이상의 센서가 감지될 경우 가중치는 상위 센서의 가중치를 1로 두고 나머지 센서의 가중치를 구한 후에 정규화하여 구할 수 있다. 예를 들어 0번과 1번 센서가 감지되면 w<sub>f</sub> = 0 이 되며 w<sub>0</sub> = 1로 두고 w<sub>1</sub>을 구한 후에 정규화 하여 구한다. 이때 센서의 우선 순위는 (0 → 1 → 7 → 6 → 2 → 3 → 5 → 4)로 하면 로봇의 운동은 가중치에 의한 비대칭이 되어 PFM이나 VFM에서의 국소 진동 문제를 해결 할 수 있다.

벽을 따라 목표를 탐색하는 알고리즘 또한 장애물 회피와 유사한 방법으로 구현할 수 있으며 로봇 진행각도 θ<sub>d</sub>만 달라진다. 예를 들어 그림 8 (a)를 고려하자. 그림과 같이 로봇의 우측에 벽이 위치한 경우에는 θ<sub>d</sub> = 0이며 그림 8 (b)를 보면 벽이 우측 상단에 위치하여 로봇이 이동하여야 하는 방향은 F 방향이다.

표 2에 장애물 회피와 벽을 따라가기 위한 로봇의 진행각도를 보였다. 동시에 여러 센서가 감지될 경우에 로봇 진행각도는 가중평균으로 구하며 강화 학습에 의하여 보정한다. 학습은 로봇의 위치와 자세에 의하여 상(reward)과 +벌(+penalty), -벌(-penalty)을 부과하고

상일 때 로봇진행방향은 표 2의 각도들에 대한 가중-합으로 하고 +벌일 때에는 가중-합에  $\Delta\theta$  를 더하며 -벌일 때에는 가중-합에서  $\Delta\theta$  를 뺀다.

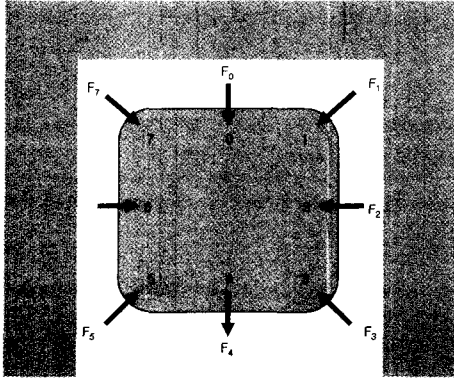
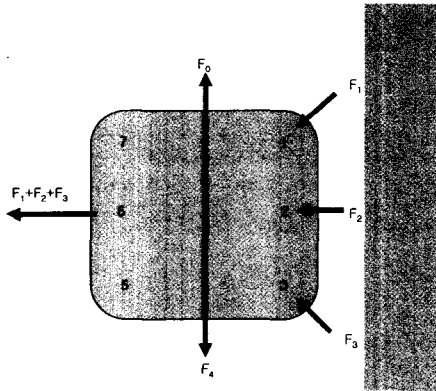
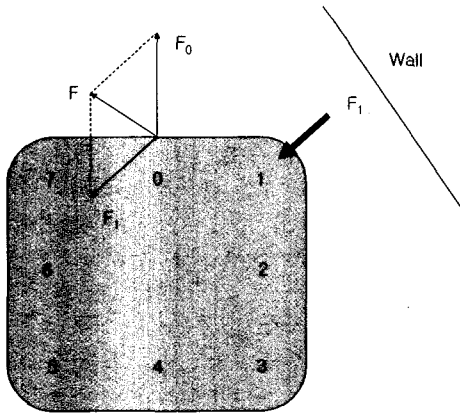


그림 7. 막다른 곳에서 탈출  
Fig. 7. Escape from concave obstacle



(a) 벽에 평행한 경우



(b) 벽이 대각선상에 있을 경우

그림 8. 장애물 회피 과정  
Fig. 8. Obstacle avoidance procedure

표 2. 센서에 따른 로봇 진행 각도  
Table 2. Angle of robot movement for detected sensor

sensor	Obstacle avoidance	Wall following
0	$-\pi$	$\pi/2$
1	$\pi/4$	$\pi/4$
2	$\pi/6$	0
3	$\pi/9$	$-\pi/6$
4	0	$\pi/2$
5	$-\pi/9$	$\pi/6$
6	$-\pi/6$	0
7	$-\pi/4$	$-\pi/4$

(3) 행동선택 부-에이전트

각 계층의 행동은 행동 선택 에이전트에 입력되어 수행성능, 학습 또는 우선순위 규칙에 의하여 행동이 선택되게 된다. 본 논문에서는 상위계층의 행동을 우선으로 하지만 수행 성능을 포함한 행동 선택을 하도록 하였다. 즉, 상위계층의 행동과 하위 계층의 행동이 서로 상충될 경우 하위계층의 행동은 무시되지만 수행성능 평가지수가 하위계층의 행동이 높으면 하위계층의 행동을 선택하도록 한다. 그림 9는 장애물 회피 과정을 선택하지 않은 과정일 지라도 장애물을 회피 할 수 있음을 보인다.

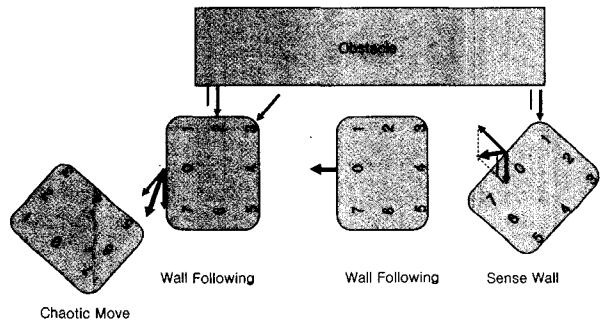


그림 9. 벽면에서의 장애물 회피  
Fig. 9. Obstacle avoidance at wall

그림12에서 1번 센서에 신호가 감지되면 로봇의 행동은 상위 계층인  $F_0+F_1$ 이 선택되어 진행하며 1, 2, 3번 센서가 감지되게 된다. 이 경우 센서는 (□) → (1) → (2) → (3)의 순서로 변화하며 (3)에서 장애물 회피 행동으로 바뀌어야 한다.

벽을 따라가는 행동과 장애물 회피 행동은 대부분의 경우 서로 충돌하며 상위계층인 벽을 따라가는 행동만이 선택되게 된다. 이러한 경우 초기에는 벽을 따라가는 행동이 선택되어 진행 되지만 물체가 장애물일 경우 센서의 신호가 변화하게 되며 이러한 경우에는 하위 계층의 행동을 선택하도록 하여야 한다. 이 과정을 그림 10에 보였다.

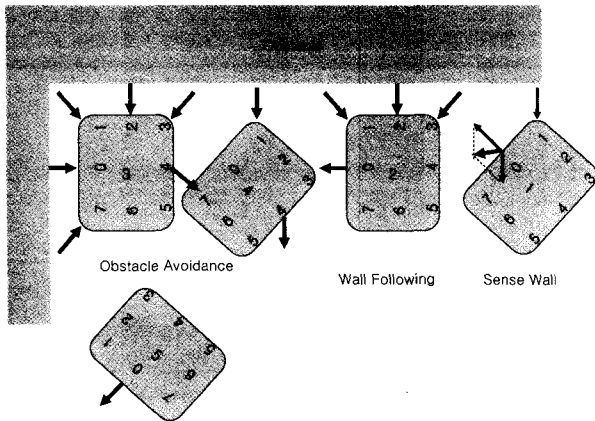


그림 10. 코너에서의 장애물 회피  
Fig. 10. Obstacle avoidance at coner

그림 10은 벽을 따라가다 코너에 들어 갔을 경우이다. 이와 같이 로봇이 코너에 들어 갔을 경우에는 (7, 0, 1) 센서 또는 (3, 4, 5)가 동시에 1이 되며 벽을 따라 가다가 이 두 경우에는 하위 계층인 장애물 회피로 전환 하여야 한다. 따라서 장애물 회피를 결정 할 때에는 수행성능 값을 크게 하여 행동선택에서 이 행동을 선택할 수 있도록 해야 한다. 그림 11에 로봇의 행동 선택 전체에 대한 유사코드를 보였다.

#### IV. 모의실험 및 실험

##### 4.1. 모의실험 1

그림 12는 제시된 장애물 회피 알고리즘에 대한 모의실험 결과이다. 모의실험에서 장애물은 중심이 (50,50), (70,100), (120,10)에 위치하며 가로-세로의 길이가 각각 20 cm인 상자로 선택하였다. Chaose한 로봇의 거동을 얻어내기 위하여 로봇의 바퀴 속도는 다음 식으로 구하였다.

$$V_r = 60 + 20\cos(4\pi(\text{rand}(1) - 0.5))$$

$$V_l = 60 + 20\sin(4\pi(\text{rand}(1) - 0.5))$$

```

Function obstacle avoidance
Input ← sensor signal
Output ← velocity, performance value
Force ← sensor signal
Net force ← Sum forces
Velocity1 ← Net force
Performance ← sensor signal

Function Wall Following
Input ← sensor signal, previous velocity
Output ← velocity
Force ← sensor signal
Net force ← Sum forces+ previous force
Velocity2 ← Net force

Function Goal
Input ← sensor signal for goal(bottom sensor)
Output ← 1, or 0
0 ← found goal
    
```

```

1 ← not found goal

Function Action Selection
Input ← performance, velocity1, velocity2, goal
Output ← Action velocity
Action velocity = (goal performance) * velocity1
← Performance = 1
Action velocity = (goal performance) * velocity2
← Performance = 0
    
```

그림 11. 로봇제어 유사코드

Fig. 11. Pseudo code for robot control

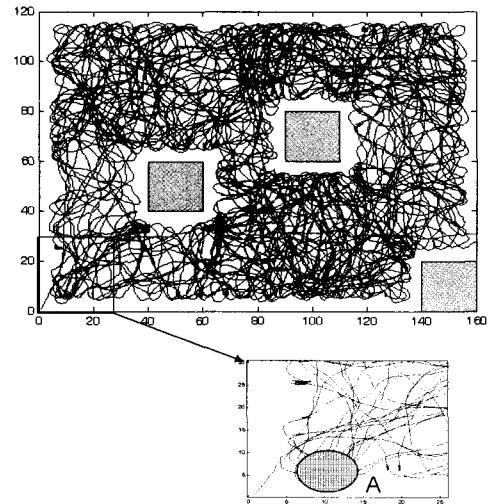


그림 12. 탐색 결과 (a)

Fig. 12. Exploration Result (a)

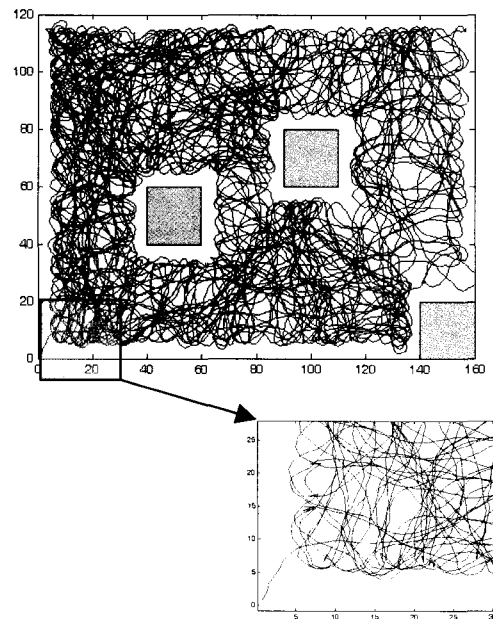
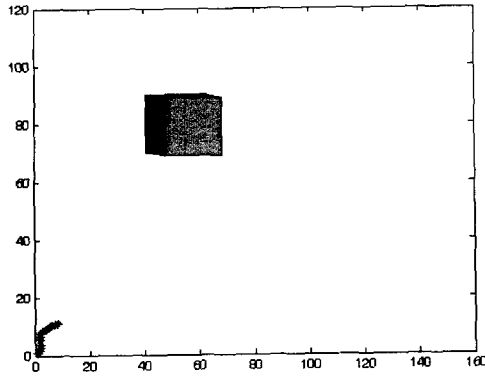


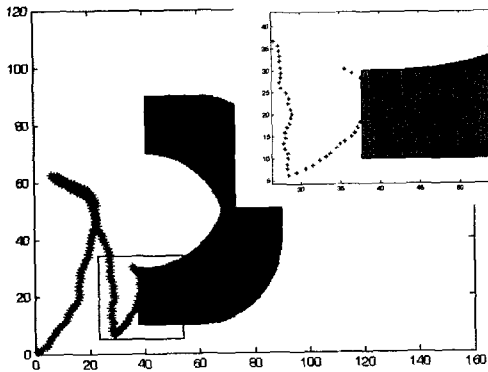
그림 13. 탐색 결과 (b)

Fig. 13. Exploration result (b)

그림에서 보여지는 바와 같이 로봇은 장애물과 벽을 잘 회피하고 있음을 알 수 있다. 그림의 A영역은 로봇이 정면으로 아래쪽 벽(y=0축)을 향하여 진행한다



(a) 초기상태



(b) 움직이는 장애물에 대한 회피

그림 14. 움직이는 장애물에 대한 회피

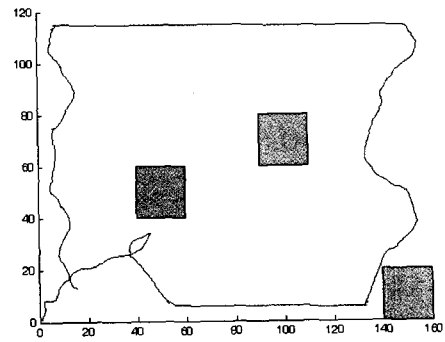
Fig. 14. Obstacle avoidance with moving object

정면의 센서가 감지되어 후진하고 센서가 감지되는 영역을 벗어나면 다시 전진하는 과정을 반복하게 되어 나타나는 현상이다. 그림 13은 그림 12의 진동현상을 제거하기 위하여 로봇이 후진하여야 할 때 후진과 회전을 동시에 하도록 하였을 때의 모의실험 결과이다. 그림 12와는 다르게 로봇이 전진-후진을 반복하는 진동현상이 없어짐을 알 수 있다.

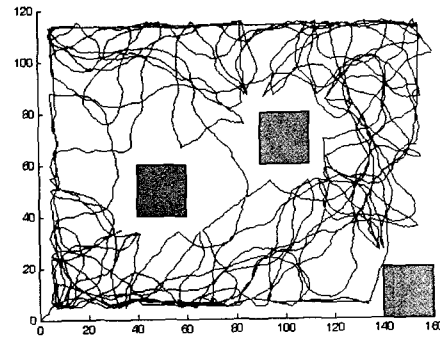
그림 14는 움직이는 물체를 장애물로 인식하고 회피하는 과정을 보여준다. 그림 (a)는 로봇이 움직이기 시작하는 초기상태이며 그림 (b)는 이동 중에 움직이는 장애물과 만났을 때의 장애물 회피 과정이다. 그림 15은 벽 따라가기와 장애물 회피에 대한 모의실험 결과이다. 그림 12와 16의 chaotic한 탐색 보다는 벽면을 따라 진행하려는 경향을 보여주고 있다.

#### 4.2. 모의실험 2

모의실험 2에서는 알려져 있지 않으며 이동하는 물체가 있는 영역을 탐색하기 위한 실험으로 2대의 로봇을 이용



(a) 초기상태



(b) 탐색결과

그림 15. 탐색결과: 벽 따라가기와 장애물 회피

Fig. 15. Exploration result: wall following and obstacle avoidance

하여 실험을 수행하였다. 행동 목표는 장애물 회피와 영역 탐색을 위한 chaotic 거동을 목표로 한다. 그림 16.a는 모의실험 환경의 초기 단계이며 b는 장애물 회피 거동이다. 장애물은 각각 가로 세로가 10 cm인 상자로 설정하였으며 b의 오른쪽 그림은 왼쪽의 상자 안을 확대하였다. 그림이며 A, B, C지점에서 장애물을 회피하고 있음을 보여준다. 그림 17.a는 센서가 감지 되어 검출된 위치를 평면상에 나타내었으며 b는 이를 3차원으로 나타내었다. c는 b에서 잡음성분을 제거한 그림이다. 그림에서 보여지는 바와 같이 벽과 고정 장애물을 잘 검출하고 있음을 알 수 있다. 또한 그림의 우측 하단에서와 같이 움직이는 물체와 그 물체의 운동 궤적도 잘 검출해 내고 있다. 그림 18은 두 대의 로봇이 이동한 전체 궤적이다.

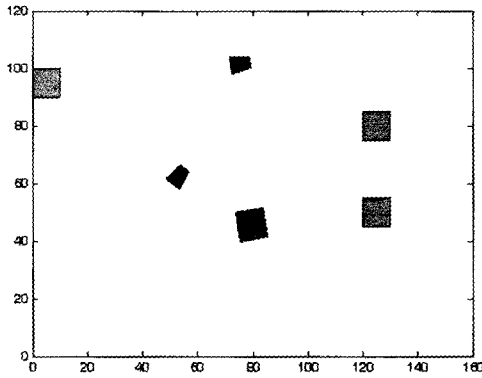
#### 4.3. 실험

실험에서 사용한 로봇은 differential drive 형태의 이동 로봇으로 제작하였다. 8개의 측면 적외선 센서와 목표를 판별하기 위한 4개의 적외선 센서를 바닥을 향하여 부착하였으며 제어용 마이크로 콘트롤러로는 Atmel사의 AT89C2051을 사용하였다. 그림 19는 실험 결과이다. 그림에서 보여지는 바와 같이 벽을 향하여 진행하던 로봇이{(1)} 벽을 인식하고 회피하려는 방향으로{(2), (3), (4)} 동작함을 알 수 있다.

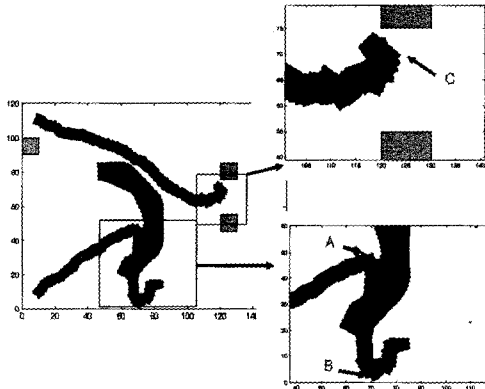


### V. 결 론

본 논문에서는 지능형 에이전트를 기반으로 하는 로봇 제어구조와 다중로봇에 대한 구조를 제시하였으며 장애물 회피와 chaotic한 탐색을 하도록 하였다. 제시된 로봇은 여러 부-에이전트들로 구성된 다중에이전트 시스템이며 각각의 부 에이전트들도 기본 동작 계층인 여러  $\mu$ -에이전트들로 구성된다. 로봇의 지능과 학습 기능은  $\mu$ -에이전트들과 여러 부-에이전트들이 학습 기능과 지능을 가지도록 하여 기존의 지능 로봇보다 학습을 쉽게 할 수 있다.

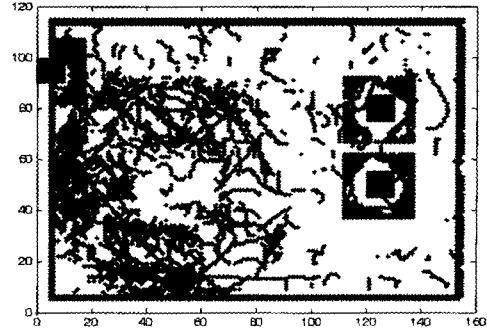


(a) 모의실험환경

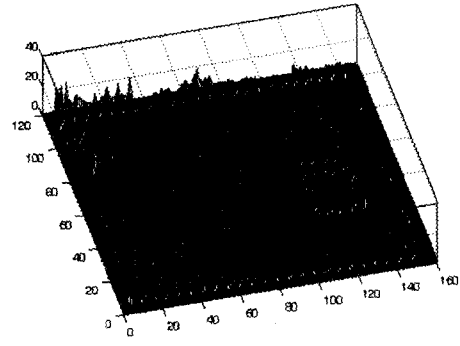


(b) 장애물 회피 거동

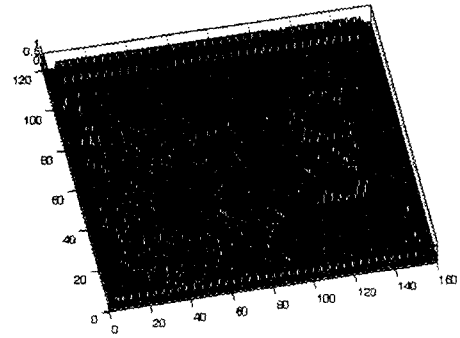
그림 16. 모의실험 환경 및 장애물 회피거동  
Fig. 16. Simulation environment and obstacle avoidance



(a) 장애물이 검출된 위치



(b) 검출된 위치의 3차원 표현



(c) 필터링된 3차원 표현

그림 17. 검출된 위치

Fig. 17. Representation of detected points

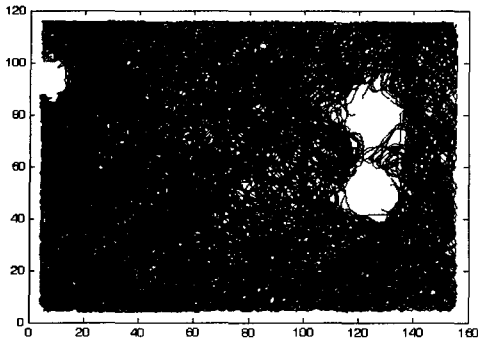


그림 18. 로봇 이동경로  
Fig. 18. Path of two robot

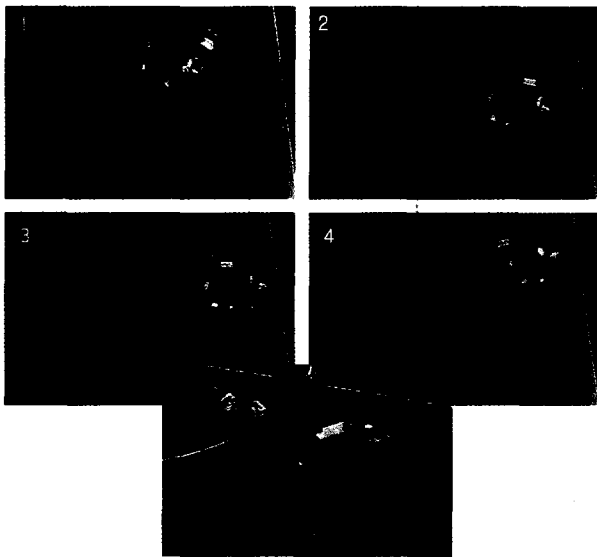


그림 19. 실험결과(위)와 제작된 로봇(아래)  
Fig. 19. Experimental result(up) and fabricated robot(bottom)

### Reference

- [1] R. Brooks. A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, RA-2, pp 14-23, April 1986.
- [2] R. C. Arkin, *Behavior-Based Robotics*, Cambridge, MA: MIT Press, 1998.
- [3] M. J. Mataric. Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8(3):304-312, June 1992.
- [4] Andrews, J. R. and Hogan, N., "Impedance Control as a Framework for Implementing Obstacle Avoidance in a Manipulator." *Control of Manufacturing Processes and Robotic Systems*, Eds. Hardt, D. E. and Book, W., ASME, Boston, 1983, pp. 243-251.
- [5] Borenstein, J. and Koren, Y., "Real-time Obstacle Avoidance for Fast Mobile Robots." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, Sept/Oct, 1989, pp. 1179-1187.
- [6] Borenstein, J. and Koren, Y., "Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments." *The 1990 IEEE International Conference on Robotics and Automation*, Cincinnati, Ohio, May 13-18, 1990, pp. 572-577.
- [7] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." *1985 IEEE International Conference on Robotics and Automation*, St. Louis, Missouri, March 25-28, 1990, pp. 500-505.
- [8] Tilove, R. B., "Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials." *General Motors Research Laboratories, Research Publication GMR-6650*, September 1989.
- [9] Y. Koren, and J. Borenstein, Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation, *Proceedings of the IEEE Conference on Robotics and Automation*, Sacramento, California, April 7-12, 1991, pp. 1398-1404
- [10] R. C. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior", *Proc. IEEE International Conference on Robotics and Automation (ICRA '87)*, Raleigh, NC, 264-271.
- [11] M. S. Atkin, G. W. King, D. L. Westbrook, B. Heeringa, A. Hannon, and P. Cohen. Spt: Hierarchical agent control: a framework for defining agent behavior. In *Proc., Intl. Conf. on Autonomous Agents*, pages 425-432, May 2001.
- [12] R. P. Bonasso, R. J. Firby, E. Gat, D. K. D. Miller, and M. Slack. Experiences with an architecture for intelligent, reactive systems. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2-3):237-256, 1997.
- [13] M. N. Nicolescu and M. J. Mataric. Learning and interacting in human-robot domains. In C. C. White and K. Dautenhahn, editors, *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans, Special Issue on Socially Intelligent Agents - The Human in the Loop*, volume 31, pages 419-430. IEEE, September 2001.
- [14] J. Yen & N. Pfluger, "A fuzzy logic based extension to Payton and Rosenblatt's command fusion method for mobile robot navigation", *IEEE Transactions on Systems, Man, and Cybernetics*, 25(6), 971-978, 1995.

- [15] D. M. Lyons and M. A. Arbib. A formal model of computation for sensory-based robotics. *IEEE Transactions on Robotics and Automation*, 5(3):280-293, June 1989.
- [16] S. Russel and P. Norvig. *AI: A Modern Approach*. Prentice Hall, NJ, 1995.
- 

## 저자 소개



### 강진식(Jin-Shig Kang)

1986년 : 제주대 통신공학과 졸업

1986년 : 한양대학교 대학원

전자통신 공학과(공학석사)

1992년 : 동 대학원(공학박사)

1992~현재 : 제주대학교

통신컴퓨터공학부(부교수)

관심분야 : 퍼지제어, 지능로봇

Phone : 064) 754-3637

Email : shigkj@cheju.ac.kr