

---

# 신경망을 이용한 자율이동로봇의 이동 경로 추종

주 기 세\*

## Moving Path Following of Autonomous Mobile Robot using Neural Network

Ki-See Joo

### 요 약

생산현장이나 불확실한 환경에서 자율이동로봇의 정확한 경로 추종은 고전적 제어 알고리즘인 경우에 많은 단점을 갖고 있다.

본 논문에서는 오류 역전파 알고리즘을 기반으로 한 신경망을 이용하여 이동로봇이 바닥 위에 설치된 선을 따라갈 수 있도록 하였다. 로봇에 부착된 3 개의 센서들로부터 인식된 정보뿐만 아니라 센서들이 인식하지 못하는 영역에서도 10등분된 세밀한 정보가 입력패턴으로 학습되기 때문에 센서들이 인식하지 못하는 영역에서도 이동로봇은 라인을 따라 원활하게 이동한다. 로봇이 목적지까지 이동하는데 걸리는 시간이 단축되고 라인과의 오차를 최소화하는 효과를 가져온다.

제안된 신경회로망 제어기의 효과를 검증하기 위하여 이동로봇의 이동 각의 변화에 따른 두 개의 모터의 속도 변화가 컴퓨터로 시뮬레이션 된다.

### Abstract

The exact path following of an autonomous mobile robot in a factory and an unreliable environment has many disadvantages in case of a classical control algorithm.

In this paper, a neural network control approach based on an error back propagation algorithm is proposed for controlling a mobile robot to follow a line installed on the road. Since not only the three recognized informations from three sensors attached on a mobile robot but also the ten detailed informations in non recognition area are learned with input patterns, a mobile robot moves smoothly an installed line in spite of

---

\* 목포해양대학교

접수일자 : 2000년 6월 9일

non perception space. The mobile robot has an effect of error minimization with a short time till a destination.

To test an effectiveness of the proposed controller, the two motor velocity changes which is affected from a moving angle change of a mobile robot are simulated with computer.

## I. 서 론

오늘날 공장 자동화를 위하여 공장내의 물류자동화에 큰 관심이 대두되고 있다. 물류 자동화를 하는데 사용된 반송장치는 컨베이어, 자율이동로봇, 스택커 크레인, 갠트리 크레인, 전동 모노레일, 로봇 등을 들 수 있으며 최근에는 자율이동로봇이 많은 비중을 차지하고 있다. 자율이동로봇은 자연 환경과 미지의 환경에서 자율적으로 이동하기 위하여 여러 감지 장치를 사용한다. 자율이동로봇을 제어하는 궁극적인 목적은 로봇이 인간의 행동을 대신할 수 있는 능력을 갖도록 하는 것이다. 자율이동로봇이 자율적으로 이동하기 위해서는 로봇의 위치, 방향, 속도제어와 같은 정확한 정보가 필요하다. 그러나 로봇의 제어 변수는 외부환경 요인 즉, 모터의 부하, 바퀴의 미끄러짐 현상, 센서의 부정확한 감지 등의 영향을 받고 있어서 자율이동로봇의 완벽한 이동은 힘들다.

지금까지 여러 감지장치를 사용한 자율이동로봇의 최적 이동경로 추종에 관한 연구가 진행되어 왔다. Janet JA, Gutierrez R. Chase TA, White MW, Sutton JC<sup>[1]</sup>는 코호넨 신경망을 사용하여 로봇의 위치를 파악하였다. 그리고 Li W, Ma CY, Wahl FM<sup>[2]</sup>은 뉴럴-퍼지 시스템을 이용하여 불확실한 환경내에서 이동로봇의 이동을 제어하였다. Loulin Huang, Jagannathan Kanniah<sup>[3]</sup>는 로봇의 몸체에 부착된 두 개의 엔코더 값을 읽으므로써 로봇의 위치를 파악하도록 하였다. 그리고 주행전략은 시행착오법을 이용하여 센서로부터 얻어진 근접한 정보를 이용하여 로봇이 이동하도록 하였다. Hans-Joachim von der Hardt, Didier Wolf Rene Husson<sup>[4]</sup>은 로봇의 위치 측정을 마그네틱 컴퍼스와 자이로스코프를 사용하여 향상시켰으며, 로봇상태의 동작을 칼만 필터(kalman filter(=EKF))의 확장에 의해 제어되도록 하였다. 여기서 EKF 알고

리즘의 실행은 로봇의 충돌에러의 방지와 불확실한 에러의 증가를 무효화하도록 하였다.

본 논문에서는 신경망을 이용하여 불확실하고, 복잡한 환경에서 로봇의 위치와 방향을 제어하여 로봇의 속도와 경로를 최적화 하도록 하였다. 기존의 PID(proportional integral derivative)제어기<sup>[5][6]</sup>를 사용한 이동로봇은 센서입력에 대한 모터의 회전과 모터에 부착된 엔코더 값의 오차를 최소화하여 라인을 추종하도록 하였다. 하지만 이 제어기를 사용하면 기준선이 3개의 센서 사이에 왔을 때는 이에 대한 제어가 불가능하다. 이러한 고전적인 제어시스템인 PID 제어기는 제어 대상에 대한 수학적인 기초 모델을 필요로하며, 이 수학적 모델은 선형적인 집합 또는 비선형 미분/적분 등식으로 이루어질 뿐만 아니라, 복잡하고 불확실한 경우 수학적으로 모델링하기가 어렵다. 그러나 학습제어시스템은 기초 모델이 없고, 불확실하고 복잡한 것을 취급할 때 성공적이며, 체계가 없는 환경에서도 작업을 순조롭게 할 수 있다. 그리고 환경의 예측, 인식, 분류, 제어 판정의 능력을 가지고 있다. 본 논문에서는 신경망을 1986년 롬멜하트(David E Rumelhart)에 의해 만들어진 오류 역전파 알고리즘(Error Back-Propagation Algorithm)을 사용하여 로봇이 라인을 오차 없이 추종하도록 하였다.

## II. 자율이동 로봇의 구성과 기능

자율이동 로봇은 사람의 개입 없이 스스로 위치를 파악하여 목적지까지 이동할 수 있는 능력을 갖추어야한다. 이동 로봇이 스스로 위치를 파악하기 위해서는 카메라나 초음파 센서와 같은 외부 센서로부터 정보를 획득해야하고, 로봇이 목적지까지 이동하기 위해서는 로봇의 중량과 모터의 사양을 고려해야한다. 본 논문에서 사용된 자율이동 로봇은 자체 제작한 로봇으로 중앙처리장치, 센서제

어장치, 구동장치로 구성되어 있다. 중앙처리장치는 자율이동 로봇의 핵심이라고 할 수 있으며, 센서제어장치로부터 로봇 외부의 변화를 감지한 데이터를 이용하여 자율이동 로봇의 경로 추종을 최적화 한 후 로봇의 이동역할을 하는 로봇의 구동장치에 최적화 된 데이터를 보내는 장치이다. 센서제어장치는 자율이동 로봇이 경로를 계속 추종하기 위하여 3개의 근접센서를 사용하였다. 그리고 구동장치는 자율이동 로봇이 이동하는데 중요한 장치로써 외부 환경요인과 하드웨어적인 요인으로 인한 오차가 많이 발생할 수 있는 부분으로 로봇의 이동을 위해 두 개의 DC모터를 사용하였으며, 모터의 정확한 회전각을 인식하기 위하여 두 개의 DC모터에 증가형 엔코더를 부착하였다. 아래 그림 1은 이동로봇의 시스템 구성을 나타내고 있다.

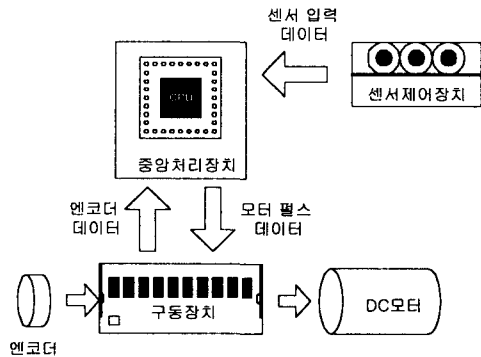


그림 1. 이동로봇 시스템의 구성  
Fig. 1 The structure of mobile robot system

### III. 이동 로봇의 경로 추종 기법

본 연구에서는 이동 로봇이 기존의 제어 시스템인 PID 제어 시스템에서 경로를 추종하는데 발생하기 쉬운 로봇의 위치측정 오차와 이에따른 로봇의 속도전락 에러를 최소화하기 위하여 학습제어 시스템인 오류 역전파 알고리즘을 사용하였다.

그림 2는 이동로봇의 구조로써 이동로봇이 바닥에 설치된 선을 추종할 때 로봇이 좌·우 변화없이 가기 위해서는 센서1만 계속적으로 인식되어야 하고 로봇의 좌·우 모터의 속도가 같아야한다. 그

런데 이동로봇이 여러 환경변수로 인하여 변화없이 가는 것은 불가능하다. 기존의 PID 제어 로봇은 바닥에 설치된 선을 추종할 때 라인이 3개의 센서에 인식되어야만 로봇의 제어가 가능하였다. 로봇이 왼쪽으로 변화하면 센서3이 인식되면서 로봇의 왼쪽모터가 오른쪽 모터에 비해서 빠르게 회전하면서 로봇은 오른쪽으로 회전하게 된다. 그리고 로봇이동이 오른쪽으로 변화하면 센서2가 인식되면서 로봇의 오른쪽 모터가 왼쪽 모터에 비해서 빠르게 회전하면서 로봇은 왼쪽으로 회전하게 된다. 본 연구에서는 이러한 센서가 인식하지 못하는 영역에서 로봇을 제어하기 위하여 오류 역전파 알고리즘을 사용하여 로봇을 제어하였다.

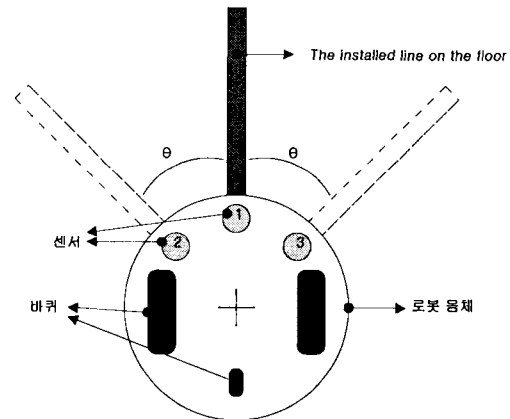


그림 2. 이동로봇의 구조  
Fig. 2 Mobile robot structure

#### 3.1 오류 역전파 알고리즘(Error Back-Propagation Algorithm)

오류 역전파 알고리즘<sup>17)</sup>은 간단하고 선형분리문제 뿐만아니라 여러 가지 문제점들을 해결할 수 있기 때문에 가장 공통적으로 사용하는 학습알고리즘으로 신경망 입력층의 각 유닛에 입력패턴을 주면 이 신호는 각 유닛에서 변환되어 은닉층에 전달되고 최후에 출력층에서 신호를 출력하게 된다. 이 출력과 목표출력과의 차이에 미분계수를 곱하여 출력 유닛에 대한 오차를 계산한다. 그리고 오차신호가 네트워크의 역방향으로 전파되어 가면

서 각각 유닛의 오차 신호가 계산되어 이것을 바탕으로 연결강도를 수정한다. 오류역전파 알고리즘의 학습과정은 다음과 같다.

1단계로 네트워크의 상태를 결정하는 연결강도  $W_{ji}$  (i번째 입력층 유닛으로부터 j번째 은닉층 유닛 간 연결강도),  $W_{kj}$  (j번째 은닉층 유닛으로부터 k번째 출력층 유닛간의 연결강도)를 각각 아주 작은 값의 임의의 수로 초기화 한다. 일반적으로 -0.5~0.5사이의 값을 사용한다.

2단계는 학습패턴을 설정한다. 본 연구에서 사용한 학습패턴은 먼저 바다에 그려진 라인 인식 유무를 나타내는  $S_j$  변수를 사용하였는데 그림 2에서 보는 바와 같이  $S_1$ 은 센서 2,  $S_2$ 은 센서 1,  $S_3$ 은 센서 3이 라인을 인식하는 경우를 각각 나타내며,  $S_4$ 는 라인이 센서 2와 1, 또는 1과 3사이에 위치하여 인식하지 못하는 경우를 나타낸다. 그리고 본 논문에서는 변수  $S_4$ , 즉 센서가 라인을 인식하지 못하는 영역(- $\theta$  ~ + $\theta$ )을 10등분( $\theta_i$ ,  $i=1, 2, \dots, 10$ )하여 학습패턴의 수를 13개로 하였다.

3단계는 학습패턴의 값을 입력층 유닛에 입력하여 출력되는 값  $O_{ji}$  와 입력층과 은닉층 사이의 연결강도  $W_{ji}$ 를 이용하여 식 (1)에서와 같이 은닉층 유닛 j의 입력  $net_{ji}$ 를 구한다. 다음으로  $net_{ji}$ 와 시그모이드 함수를 이용하여 은닉층 유닛 j의 출력  $O_{ji}$ 를 식 (2)를 이용하여 구한다.

$$net_{ji} = \sum_i W_{ji} O_{pi} \dots\dots\dots (1)$$

$$O_{ji} = f_j(net_{ji}) \dots\dots\dots (2)$$

여기서  $net_{ji}$  : 패턴 p에 대한 은닉층 유닛 j의 입력

$W_{ji}$  : i번째 입력층 유닛으로부터 j번째 은닉층 유닛간의 연결 강도

$O_{ji}$  : 패턴 p에 대한 은닉층 유닛 j의 출력

4단계는 은닉층 유닛의 출력  $O_{ji}$ 와 은닉층과 출력층 사이의 연결강도  $W_{kj}$ 를 이용하여 출력층 유닛 k의 입력  $net_{pk}$ 를 식 (3)을 이용하여 구한다. 다음  $net_{pk}$ 와 시그모이드 함수를 이용하여 출력층 유닛 k의 출력  $O_{pk}$ 를 식 (4)를 이용하여 구한다.

$$net_{pk} = \sum_j W_{kj} O_{pj} \dots\dots\dots (3)$$

$$O_{pk} = f_k(net_{pk}) \dots\dots\dots (4)$$

여기서  $net_{pk}$ : 패턴 p에 대한 출력층 유닛 k의 입력

$W_{kj}$ : j번째 은닉층 유닛으로부터 k번째 출력층 유닛간의 연결강도

$O_{pk}$ : 패턴 p에 대한 출력층 유닛 k의 출력

5단계는 학습패턴의 목표출력  $t_{pk}$ 와 실제 출력  $O_{pk}$ 와의 차로부터 출력층 유닛 k에 연결된 연결강도와 출력층 유닛 k에 대한 오차  $\delta_{pk}$ 를 식 (6)을 이용하여 구한다. 본 연구에서 사용한 목표출력은  $V_l$ (왼쪽모터의 속도)과  $V_r$ (오른쪽모터의 속도)으로 하였다.

$$f'_k(net_{pk}) = \frac{\partial O_{pk}}{\partial net_{pk}} \dots\dots\dots (5)$$

$$\delta_{pk} = (t_{pk} - O_{pk}) f'_k(net_{pk}) = (t_{pk} - O_{pk}) O_{pk} (1 - O_{pk}) \dots\dots\dots (6)$$

여기서  $\delta_{pk}$  : 패턴 p에 대한 출력층 유닛 k에 대한 오차

$t_{pk}$  : 패턴 p에 대한 출력층 유닛 k에 대한 목표출력

$f'_k(net_{pk})$  :  $net_{pk}$ 의 미분

6단계는 오차  $\delta_{pk}$ 와 은닉층과 출력층간의 연결강도  $W_{kj}$ 와 은닉층의 출력  $net_{ji}$ 로부터 은닉층 유닛 j에 연결된 연결강도와 은닉층 유닛의 오차  $\delta_{ji}$ 를 식 (8)을 이용하여 구한다.

$$f'_j(net_{ji}) = \frac{\partial O_{ji}}{\partial net_{ji}} \dots\dots\dots (7)$$

$$\delta_{ji} = f'_j(net_{ji}) \sum_k \delta_{pk} W_{kj} = \sum_k \delta_{pk} W_{kj} O_{ji} (1 - O_{ji}) \dots (8)$$

여기서  $\delta_{ji}$ : 패턴 p에 대한 은닉층 유닛 j에 대한 오차

$f'_j(net_{ji})$ :  $net_{ji}$ 의 미분

7단계는 5단계에서 구한 출력층 유닛 k에서의 오차  $\delta_{pk}$ , 은닉층 유닛 j의 출력  $O_{ji}$ 와 정수  $\alpha$ 와의

급을 더하여 은닉층 유닛 j와 출력층 유닛 k에 연결된 연결강도  $W_{kj}$ 를 식 (9)와 같이 수정한다.

$$W_{kj} = W_{kj} + \alpha \cdot \delta_{pk} O_{pj} \dots\dots\dots(9)$$

여기서  $\alpha$ : 임의의 상수

8단계는 은닉층 유닛 j의 오차  $\delta_{pj}$ , 입력층 유닛 i의 출력  $O_{pi}$ , 정수  $\alpha$ 와의 곱을 더하여 입력층 유닛과 은닉층 유닛 j에 연결된 연결강도  $W_{ji}$ 를 식(10)과 같이 수정한다.

$$W_{ji} = W_{ji} + \alpha \cdot \delta_{pj} O_{pi} \dots\dots\dots(10)$$

9단계는 다음 패턴을 학습시킨다. 10단계는 모든 학습패턴에 대하여 전부 학습할 때까지 2단계로 되돌아간다. 11단계는 학습의 반복 횟수를 센다. 12단계는 학습의 반복 횟수가 제한 횟수보다 작으면 2단계로 되돌아간다. 마지막으로 13단계는 종료 단계이다.

그림 3은 오류 역전파 알고리즘의 순서도를 나타내고 있다.

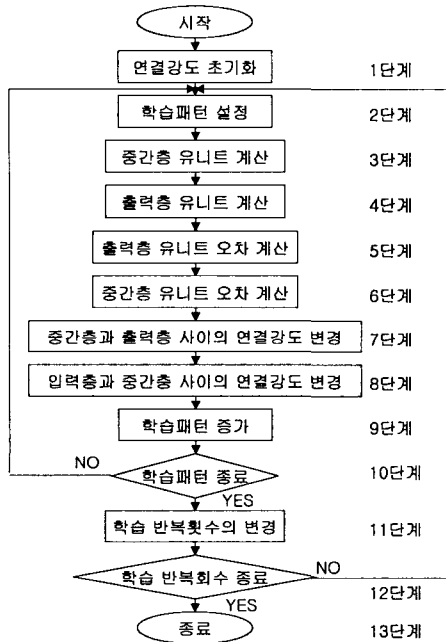


그림 3. 오류 역전파 알고리즘의 순서도  
Fig. 3 Error back-propagation algorithm flow

### 3.2 로봇 제어 시스템 구성

이동로봇을 목표지점까지 최단시간에 이동하기 위해서는 로봇이 라인으로부터 벗어나는 에러를 줄여야한다. 그러기 위해서는 로봇에 여러 센서를 부착하여 로봇이 라인을 벗어날 경우 위치를 바로 잡도록 하여야한다. 하지만 라인이 센서 사이에 위치하게 될 경우, 즉 센서가 인식하지 못하는 영역에서는 로봇의 제어가 불가능하게 된다. 본 연구에서는 오류 역전파 알고리즘을 사용한 신경망을 이용하여 이를 해결하였다. 그림 4는 신경망을 사용한 로봇 제어 시스템<sup>[8]</sup> 구성도를 나타내고 있다.

그림 4에서  $S_j$ 변수 값은 로봇에 부착된 세 개의 센서 중 어느 센서가 인식하는지의 여부에 따라 결정되는데  $S_1$ 은 왼쪽센서 인식,  $S_2$ 는 중앙센서 인식,  $S_3$ 은 오른쪽 센서가 인식하는 경우를 각각 나타낸다. 마지막으로  $S_4$ 는 세 센서가 모두 인식하지 못하는 영역인 경우를 나타낸다.

그리고  $\theta_i$ 는 센서가 인식하지 못하는 영역에서의 로봇의 위치 각을 나타내는 변수로서 값의 범위는  $-50^\circ \leq \theta_i (i= 1, 2 \dots 10) \leq +50^\circ$ 로 하였다.

그림 4에서  $\theta_i$ 는 센서가 인식하지 못하는 영역에서의 원하는 로봇의 위치 각을 나타내며,  $S_j$ 는 원하는 센서입력 값,  $\theta_0$ 는 실제 로봇의 위치 각,  $S_0$ 는 실제 센서 입력값,  $V_l$ 은 왼쪽바퀴 속도,  $V_r$ 은 오른쪽 바퀴속도를 각각 나타내며, 양쪽 바퀴의 속도는  $0\text{rpm} \leq V \leq 50\text{rpm}$ 로 하였다.

본 시스템에서 이동 로봇이 라인을 따라 이동하게 되면 로봇의 위치와 센서 값이 변하게 된다. 이때 신경회로망은 로봇이 라인으로부터 벗어나는 에러<sup>[9][10]</sup>를 줄이는 역할을 한다. 그림 4에서 신경회로망에 입력되는 변수는 로봇의 위치각( $\theta_i$ )과 센서 입력 값( $S_j$ )이고, 출력 변수는 로봇의 왼쪽 모터의 속도( $V_l$ )와 오른쪽 모터의 속도( $V_r$ )이다. 그리고 학습제어 알고리즘은 로봇의 입력 값( $\theta_i, S_j$ )과 로봇의 이동으로 인한 결과 값( $\theta_0, S_0$ )의 차이를 비교하고 오차를 계산하여 신경회로망에 입력된다. 그 결과 이동 로봇이 라인으로부터 오차가 생기게 되면 이를 보완해주는 역할을 하게 된다. 그리고 로봇의 위치 각의 추정방법은 로봇이 이동을 하면 2개의 모터가 회전을 하게 되고 모터의 회전수를 엔코더가 체크하게된다. 그리고 엔코더의 회전수의

차를 비교하여 로봇의 위치 각을 측정하게 된다.

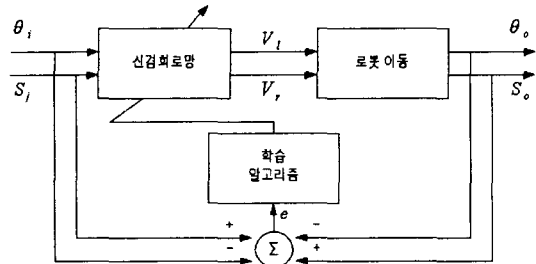


그림 4. 신경망을 사용한 로봇제어 시스템 구성도  
Fig. 4 The Structure of robot control system using neural network

### 3.3 로봇의 신경회로망 구성

그림 5는 신경망 구성도를 나타내고 있는데 신경 회로망의 입력층 유닛은 로봇에 부착된 센서 입력 값과 센서들이 인식하지 못하는 영역에서의 로봇의 위치 각으로 하였다. 로봇의 위치 각의 범위는 라인으로부터 왼쪽으로 50°, 오른쪽으로 50°로 하였으며, 센서 입력 값은 왼쪽 센서, 오른쪽 센서, 가운데 센서, 센서들이 인식하지 못함으로 하였다. 신경망의 출력층 유닛은 로봇의 왼쪽과 오른쪽 모터의 속도 값으로 하였고 왼쪽과 오른쪽 모터의 속도 값의 범위는 0부터 50까지의 펄스 값으로 하였다. 그리고 입력층과 출력층 사이의 히든층(hidden layer)은 2개로 하였으며 첫 번째 히든층과 두 번째 히든층의 신경세포의 개수를 13개로 구성하였다.

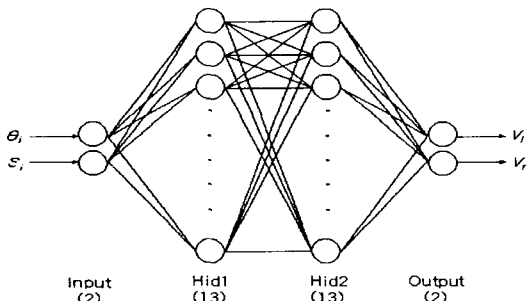


그림 5. 신경 회로망 구성도  
Fig. 5 Neural network structure

### 3.4 이동로봇의 라인 추종 전략

이동로봇을 처음으로 가동하면 가운데 센서( $S_2$ ) 값이 입력되게 된다. 그리고 로봇이 이 값을 유지하기 위하여 양쪽 모터의 속도( $V_L, V_R$ )를 같도록 하여야 한다. 만약 다른 환경변수의 영향으로 로봇이 라인과 오차가 발생하게 되면 새로운 위치 각( $\theta_0$ )과 센서입력( $S_0$ )값이 출력되게 된다. 그리고 기존의 위치 각( $\theta_i$ ), 센서 입력( $S_i$ ) 값을 새로운 위치 각( $\theta_0$ ), 센서 입력( $S_0$ ) 값과 비교하여 라인과 오차를 계산한다. 이 오차를 오류 역전파 알고리즘을 사용하여 수정된  $V_L$ 과  $V_R$ 값이 출력되게된다. 만약 이동로봇의 위치 각이 50°로 변화가 생기고 센서에 입력이 없으면 이동로봇은 오류 역전파 알고리즘에 의해서 왼쪽 모터의 속도가  $V_L=10$ 으로 되고 오른쪽 모터의 속도가  $V_R=40$ 이 된다. 아래는 이동로봇의 라인 추적 알고리즘을 나타내고 있다.

- 단계 1:  $\theta_i, S_i$ 를 입력
- 단계 2:  $V_L, V_R$ 이 출력
- 단계 3: 새로운  $\theta_0, S_0$ 가 생성
- 단계 4: 라인과 오차계산
- 단계 5: 오류 역전파 알고리즘
- 단계 6: 수정된  $V_L, V_R$ 이 출력
- 단계 7: 단계 3으로 이동

## IV. 실험 및 결과

이동로봇이 라인과 오차를 최소화하기 위하여 신경회로망<sup>[11]</sup>을 이용하여 로봇의 이동을 최적화하였다. 신경회로망을 로봇에 적용했을 경우의 컴퓨터 시뮬레이션 결과는 다음과 같다.

이동로봇이 라인의 어느 위치에 있는가에 따라 로봇의 좌우 모터의 속도를 조절해 주어야한다. 그림 6은 로봇의 왼쪽 센서가 라인을 인지하였을 경우 좌·우 모터의 시뮬레이션 결과이다. 이 경우에는 왼쪽 모터속도가 10이고 오른쪽 모터속도가 50 일 때가 로봇이 라인과 오차를 줄일 수 있는 최적화 된 값이다. 그림 6에서 보면 학습횟수가 500 회 일 때 좌우 모터속도가 원하는 값과 일치한다. 그림 7은 로봇의 가운데 센서가 라인을 인지하였을 경우 좌우 모터의 시뮬레이션 결과로써 왼쪽과 오른쪽 모터속도가 50일 때가 로봇이 오차를 없앨

수 있는 최적화 된 값이다. 이 경우는 학습횟수가 300회 일 때 좌우 모터의 속도가 원하는 값과 일치하게 된다.

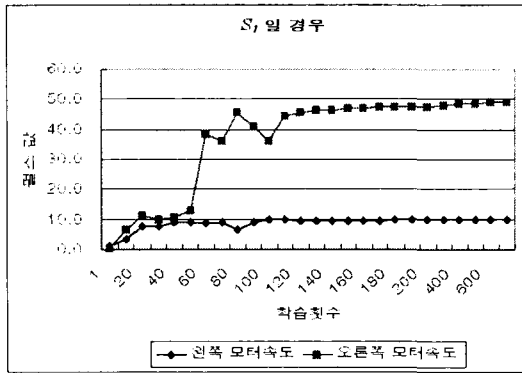


그림 6.  $S_1$  일 경우 모터속도  
Fig. 6 Motor velocity in case of  $S_1$

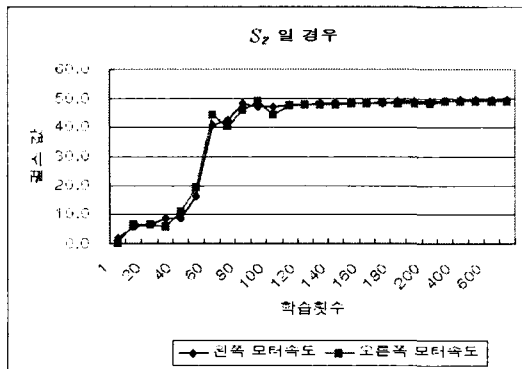


그림 7.  $S_2$  일 경우 모터속도  
Fig. 7 Motor velocity in case of  $S_2$

그림 8은 로봇의 오른쪽 센서가 라인을 인지하였을 경우 좌우 모터의 시뮬레이션 결과로써 왼쪽 모터속도가 50이고 오른쪽 모터속도가 10일 때가 오차를 줄이는데 최적화 된 값이다. 이 경우 학습횟수를 300회로하면 좌우모터의 속도가 원하는 값과 거의 일치하게 된다. 그림 9는 센서들이 인식하지 못하는 영역에서의 로봇의 위치 각이  $-50^\circ$ 일 경우 좌우 모터의 시뮬레이션 결과로써 학습횟수를 500회로하면 왼쪽 모터속도가 10이고 오른쪽 모터

속도가 40으로 라인과의 오차를 줄일 수 있는 값이 된다.

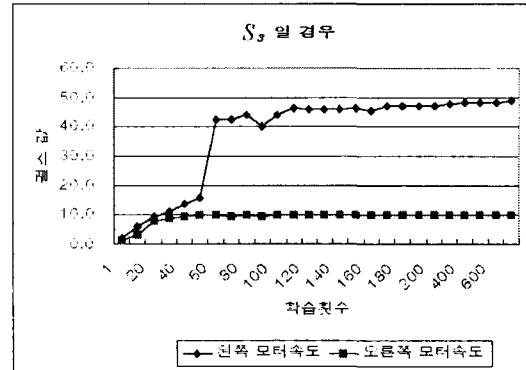


그림 8.  $S_3$  일 경우 모터속도  
Fig. 8 Motor velocity in case of  $S_3$

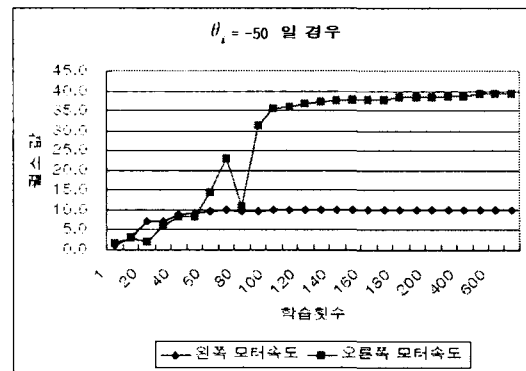


그림 9.  $\theta_i = -50$  일 경우 모터속도  
Fig. 9 Motor velocity in case of  $\theta_i = -50$

그림 10은 센서들이 인식하지 못하는 영역에서의 로봇의 위치 각이  $50^\circ$ 일 경우 좌우 모터의 시뮬레이션 결과로써 학습횟수를 400회로하면 왼쪽 모터속도가 40이고 오른쪽 모터속도가 10으로써 라인과의 오차를 줄일 수 있는 값이 된다. 이와같이 신경회로망의 학습횟수를 500회 정도로하면 로봇이 라인과의 오차를 줄일 수 있는 좌우 모터속도 값과 거의 일치하게 된다. 만약 학습횟수를 500회 이하로 하면 신경회로망의 학습시간을 줄이는 반면에 오차를 줄일 수 있는 값과 에러가 커지게 된다. 그

리고 학습횟수를 500회 이상으로 하면 오차를 줄일 수 있는 반면에 학습시간이 길어지게 된다.

그림 11은 위의 신경회로망의 컴퓨터 시뮬레이션 결과를 실제 이동로봇이 라인과의 오차를 최소화하기 위한 좌우 모터의 속도 제어값을 나타내고 있다. 이동로봇의 왼쪽 센서가 라인에 인지되면 왼쪽 모터의 속도는 10으로하고, 오른쪽 모터의 속도는 50으로 해서 라인과의 오차를 줄인다. 그리고 이동로봇의 오른쪽 센서가 라인에 인지되면 왼쪽 모터의 속도는 50이고, 오른쪽 모터의 속도는 10으로해서 라인과의 오차를 최소화한다. 또 이동로봇의 가운데 센서가 라인에 인지되면 로봇이 이상 상태를 계속유지하기 위하여 좌우모터의 속도를 50으로한다.

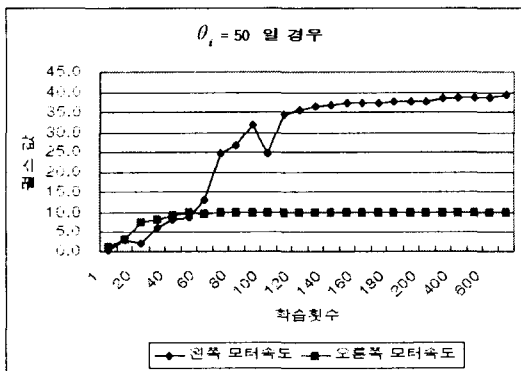


그림 10.  $\theta_i = 50$  일 경우 모터속도  
Fig. 10 Motor velocity in case of  $\theta_i = 50$

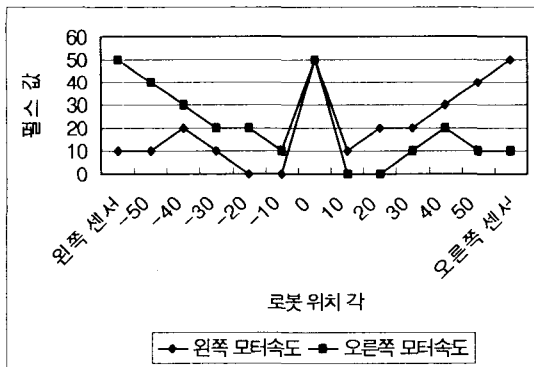


그림 11. 이동로봇의 속도제어  
Fig. 11 Mobile robot velocity control

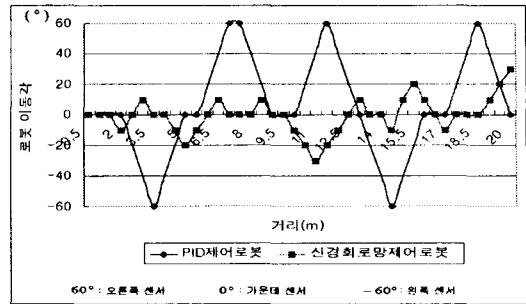


그림 12. PID제어로봇과 신경회로망 제어로봇의 이동결과 비교

Fig. 12 Moving result comparison of PID control robot & Neural network control robot

표 1. PID제어로봇과 신경회로망 제어로봇의 장단점  
Table 1. Comparison of PID control robot & Neural network control robot

	PID 제어로봇	신경회로망 제어로봇	비교
장점	<ul style="list-style-type: none"> <li>제어기 구성이 쉬움</li> <li>제어 프로그램이 간단</li> </ul>	<ul style="list-style-type: none"> <li>복잡하고 어려운 수학적 분석의 사용 소거</li> <li>난해한 비선형 제어 문제 해결</li> <li>불확실한 영역 제어 가능</li> <li>빠른 다중처리 기술</li> <li>로봇의 부드러운 이동</li> </ul>	
단점	<ul style="list-style-type: none"> <li>복잡하고 어려운 수학적 사용</li> <li>난해한 비선형 제어 문제 해결의 어려움</li> <li>불확실한 영역 제어 불가능</li> <li>로봇이동시 변화 심함</li> </ul>	<ul style="list-style-type: none"> <li>제어기 구성이 복잡</li> </ul>	

그림 12는 이동로봇이 20m를 이동 시 PID제어 로봇과 신경회로망 제어로봇의 이동결과를 나타낸 것으로 PID제어를 사용한 로봇은 라인을 인지하는 부분이 왼쪽 센서, 오른쪽 센서, 가운데 센서로서 로봇이 이동하는데 좌우변화가 심하게 발생한다. 반면에 신경회로망을 이용한 로봇은 센서가 인지하지 못하는 부분까지도 제어가 가능하므로 로



봇의 변화가 라인으로부터 거의  $\pm 20^\circ$  내에서 이동하게 되며, 목적지까지의 시간을 절약할 수 있다. 그리고 표 1은 PID제어로봇과 신경회로망 제어로봇을 비교 설명한 것으로 신경회로망 제어로봇이 PID제어로봇에 비하여 복잡하고 난해한 영역의 제어가 가능하므로 본 연구에서 이동로봇이 라인을 이동할 때 뛰어난 성능을 발휘하는 것을 확인하였다.

### V. 결론

본 연구에서는 기존의 PID제어로봇과는 달리 신경회로망을 이용하여 이동로봇을 제어함으로써 이동로봇이 라인을 이동할 때 라인과의 오차를 최소화함으로써 이동로봇이 목적지까지 이동하는 시간을 단축할 수 있도록 실험하였다. 실험결과 기존의 PID제어를 사용한 로봇보다는 신경회로망을 이용한 로봇이 라인과의 이동 각의 변화를 줄일 수 있었으며 목적지까지 이동시간을 단축할 수 있었다. 향후 연구에서는 이동로봇이 라인을 추적하면서 장애물이 있을 경우 이를 회피하는 연구<sup>[12]</sup>와 환경이 주어지지 않아도 로봇이 스스로 제어하여 이동할 수 있도록 연구가 필요하다.

### 참고문헌

[1] Janet JA, Gutierrez R, Chase TA, White MW, Sutton JC, "Autonomous Mobile Robot Global Self-Localization Using Kohonen and Region-Feature Neural Networks", Journal of Robotic Systems, Vol. 14, No. 4, in 1997, pp. 263-282.

[2] Li W, Ma CY, Wahl FM, "A Neuro-Fuzzy System Architecture for Behavior-Based Control of a Mobile Robot in Unknown Environments", Journal of Fuzzy Sets and Systems, Vol. 87, No. 2, in 1997, pp. 133-140.

[3] Loulin Huang and Jagannathan Kanniah, "An Approach for Position Estimation and Navigation of a Mobile Robot", Electronics

and Communications Department Singapore Polytechnic, 500 Dover Road, Singapore 139651 REPUBLIC OF SINGAPORE, 1996.

[4] Hans-Joachim von der Hardt, Didier Wolf Rene Husson, "Localization of a Wheeled Mobile Robot using Incremental Compass", Centre de Recherche en Automatique de Nancy (CNRS URA 821) 2, avenue de la Foret de Haye, 54516 Vandoeuvre, France, 1996.

[5] Pervozvanski, AA and Freidovich, LB, "Robust stabilization of robotic manipulators by PID controllers", Dynamics and Control, Vol. 9, No. 2, pp. 203-222, 1999.

[6] Seraji, H, "A new class of nonlinear PID controllers with robotic applications", Journal of Robotic Systems, Vol. 15, No. 3, pp. 161-181, 1998.

[7] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representation by error propagation", Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, 1986, pp. 318-362.

[8] S.H. Han, H.S. Lee, "The Design of Fuzzy-Neural Controller for Velocity and Azimuth Control of a Mobile Robot", Journal of the Korean Society of Precision Engineering, Vol. 13, No. 4, April. 1996.

[9] B.R. Lee, W.K. Lee, H.C. Yi, "Path Control of a Mobile Robot Using Fuzzy-Neural Hybrid System", Journal of the Korean Society of Precision Engineering, Vol. 12, No. 8, August, 1995.

[10] Curran A, Kyriakopoulos KJ, "Sensor-Based Self-Localization for Wheeled Mobile Robots", Journal of Robotic Systems, Vol. 12, No. 3, in 1995, pp. 163-176.

[11] R. Fierro and F. L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Networks", IEEE Transactions On Neural Networks, Vol. 9, No. 4, JULY 1998.

- [12] S. W. Kim, D. G. Gweon, Y. Y. Cha, "A Collision Avoidance Algorithm of a Mobile Robot in the Presence of Moving Obstacle", Journal of the Korean Society of Precision Engineering, Vol. 14, No. 1, JANUARY, 1997.



주 기 세(Ki-See Joo)

1988년 2월 한양대학교 공과대학 산업공학과(공학사)

1992년 5월 Texas A&M Univ. (공학석사)

1996년 2월 고려대학교 공과대학 산업공학과(공학박사)

1996년 3월~1997년 2월 대불대학교 기계·산업시스템 공학부 전임강사

1997년 3월~현재 목포해양대학교 해상운송시스템 공학부 전임강사

관심분야: 로봇 비전, 이동로봇, 데이터베이스